

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.041

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

“ ” _____ 2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: Об'єктно-орієнтована модель системи управління телекомунікаційною мережею

Виконав: студент другого курсу, групи ІТ-74мп
(шифр групи)

_____ Мошенченко Микита Сергійович _____

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник професор, д.т.н. Жураковський Б.Ю. _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант _____

(назва розділу)

_____ (науковий ступінь, вчене звання, прізвище, ініціали)

_____ (підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2018 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

І.Р. Пархомей

(підпис)

«__» _____ 2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Мошенченко Микиті Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації «Об'єктно-орієнтована модель системи управління телекомунікаційною мережею», _____

науковий керівник дисертації професор, д.т.н. Жураковський Б.Ю. _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «07» листопада 2018 р. №4112-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження – процес управління телекомунікаційною мережею.

4. Предмет дослідження – управління та моніторинг телекомунікаційної мережі, надання дружелюбного інтерфейсу для користувача.

5. Перелік завдань, які потрібно розробити – аналіз проблеми та існуючих рішень; розробка моделі; дослідження ефективності розробленої моделі.

6. Орієнтовний перелік ілюстративного матеріалу – шість плакатів

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 03.09.2018 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	13.09.2018 р.	
2	Постановка задачі	15.09.2018 р.	
3	Аналіз інформаційного забезпечення	20.09.2018 р.	
5	Аналіз алгоритмічного забезпечення	25.09.2018 р.	
6	Розробка алгоритмічного забезпечення	15.10.2018 р.	
7	Розробка моделі	01.11.2018 р.	
8	Маркетинговий аналіз стартап-проекту	10.11.2018 р.	
9	Висновки	15.11.2018 р.	

Студент

(підпис)

Мошенченко М.С.
(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

Жураковський Б.Ю.
(ініціали, прізвище)

АНОТАЦІЯ

У роботі розглянуто проблему процесу управління телекомунікаційною мережею, показано основні особливості існуючих рішень проблеми, їх переваги та недоліки.

Розроблено об'єктно-орієнтовану модель системи управління, що надає можливість більш ефективно керувати мережею. Дана модель може бути використана на підприємствах, що надають телекомунікаційні послуги. Дана розробка дозволяє зменшити навантаження при роботі та зекономити ресурси підприємства.

Ключові слова: мережа NGN, модульність, телекомунікаційна мережа.

Розмір пояснювальної записки – 113 аркушів, містить 33 ілюстрації, 31 таблицю, 6 додатків.

ABSTRACT

Examines the problem of telecommunication network management process is considered, the main features of the existing solutions of the problem, their advantages and disadvantages are shown.

Developed the object-oriented model of the control system is developed, which provides an opportunity to more effectively manage the network. This model can be used at enterprises providing telecommunication services. This development allows you to reduce workload and save enterprise resources.

Keywords: NGN network, modularity, telecommunication network..

The size of the explanatory note is 113 pages, contains 33 illustrations, 31 tables, 6 appendixes.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	9
ВСТУП.....	10
1. ЗАГАЛЬНИЙ ОГЛЯД СИСТЕМ УПРАВЛІННЯ.....	12
1.1 Поняття системи управління.....	12
1.2 Загальні поняття про мережу управління телекомунікаціями	13
1.3 Стандарти систем управління	22
1.4 Реалізація систем управління мережами та послугами телекомунікацій	24
1.4.1 Система управління мережами NGN	26
1.4.2 Система управління мережами SDH	31
1.4.3 Система управління телефонними мережами	32
1.4.4 Система управління трафіком	36
Висновки до розділу.....	38
2. РОЗРОБКА ОБ'ЄКТНО-ОРІЄНТОВАНОЇ МОДЕЛІ СИСТЕМИ УПРАВЛІННЯ ТЕЛЕКОМУНІКАЦІЙНОЮ МЕРЕЖЕЮ	39
2.1 Постановка задачі.....	39
2.2 Введення в об'єктно-орієнтовану методологію розробки систем	40
2.3 Аналіз вимог і визначення специфікацій програмного забезпечення при об'єктному підході.....	51
2.4 Розширення функціональних можливостей	54
2.5 Порядок виконання процесів	58
2.6 Інформаційна модель	61
Висновки до розділу.....	64

3. ПОБУДОВА ОБ'ЄКТНО-ОРІЄНТОВАНОЇ МОДЕЛІ СИСТЕМИ УПРАВЛІННЯ ТЕЛЕКОМУНІКАЦІЙНОЮ МЕРЕЖЕЮ	66
3.1 Побудова концептуальної моделі предметної області	66
3.2 Динамічні, статичні моделі та моделі реалізації об'єктно-орієнтованих програмних систем.....	69
3.3 Розробка структури програмного забезпечення при об'єктному підході.....	70
3.4 Канали управління	74
3.5 Моделі процесів.....	84
3.6 Об'єктно-орієнтована модель системи управління мережею	90
3.7 Об'єктно-орієнтована модель системи управління в основних 4-х станах ..	95
Висновки до розділу.....	101
4. РОЗРОБКА СТАРТАП-ПРОЕКТУ.....	102
4.1 Опис ідеї проекту	102
4.1.1 Уявлення про зміст ідеї.....	102
4.1.2 Аналіз потенційних техніко-економічних переваг ідеї.....	102
4.2 Технологічний аудит ідеї проекту.....	102
4.3 Аналіз ринкових можливостей запуску стартап-проєкту	103
4.3.1 Аналіз попиту	103
4.3.2 Аналіз груп клієнтів	103
4.3.3 Аналіз загроз та можливостей	104
4.3.4 Аналіз конкуренції	104
4.3.5 Аналіз пропозиції	105
4.3.6 Аналіз та обґрунтування факторів конкурентоспроможності.....	105
4.3.7 SWOT - аналіз	106
4.4 Розроблення ринкової стратегії проекту.....	107

4.4.1 Розроблення ринкової стратегії проекту.....	107
4.4.2 Формування стратегії розвитку та визначення стратегії конкурента ...	108
4.5 Розроблення маркетингової програми стартап-проекту	109
4.5.1 Формування маркетингової концепції та моделі	109
4.5.2 Формування системи збуту та розроблення комунікацій	109
Висновки до розділу.....	110
ЗАГАЛЬНІ ВИСНОВКИ	111
ПЕРЕЛІК ПОСИЛАНЬ	112
ДОДАТКИ.....	113

ПЕРЕЛІК СКОРОЧЕНЬ

СУ - система управління;

ООА - об'єктно-орієнтований аналіз;

ООМ - об'єктно-орієнтована методологія;

ПЗ – програмне забезпечення;

ДПДД - діаграма потоків даних дій;

ВПУ - вузловий пункт управління;

ДПС - діаграма переходу в стан;

ТПС - таблиця переходу в стани;

ЗЛ – з'єднувальна лінія;

ОУ – об'єкт управління

ВСТУП

Швидкий розвиток технологій телекомунікацій у світі ставить перед нашою державою такі завдання, як надати користувачам широкий вибір послуг у сфері інформації на високому, тобто, сучасному рівні. Телекомунікаційна мережа є фундаментом величезного комплексу, який включає в себе багато варіантів передачі, управління та обробки інформації. Щоб досягти такої глобальної доступності для користувачів, реалізувати вимоги ринку, необхідно знайти архітектуру мережі, яка хоча б змогла поєднати устаткування, яке вже знаходиться у роботі з новітніми технологіями. Такі вимоги посприяли створенню глобальної інформаційної інфраструктури, платформою якої є концепція інтелектуальних мереж (ІМ).

Системою управління називають певний набір засобів, що впливають на контролюємий об'єкт для того, щоб досягти поставленої мети за допомогою цього об'єкту. Таким об'єктом може бути як техніка так і людина. Один об'єкт може складатись з інших об'єктів, які, в свою чергу, можуть мати структуру взаємозв'язків.

Задачею управління є гарантована безперебійна та якісна служба мережі та її засобів в умовах постійного розвитку і вдосконалення. Будь-яка мережа є складним об'єктом, що має потребу бути розподіленим задля спрощення управління. Управління - це багатофункціональний процес.

У світі існує багато стандартів систем управління, проте популярними можна виділити дві: це стандарти Internet, що керують на основі протоколу SNMP (*Simple Network Management Protocol*) та CMIP (*Common Management Information Protocol*) на які спираються міжнародні стандарти OSI, розроблені ISO і ITU-T. Перша серія стандартів мінімально специфікує аспекти і елементи управління, в той час як друга серія навпаки – максимально. У сучасному світі так склалося, що корпоративні та локальні мережі для себе використовують SNMP, а ISO/ITU та

СМІР використовують якраз таки для управління мережами та послугами телекомунікацій.

Повертаючись у минуле можна зазначити, що в СРСР мали місце плани по створенню такої автоматизованої системи зв'язу, яка б була єдина для всього світу. І при розробці цієї системи значна кількість уваги приділялась якраз задачам розробки і впровадженню автоматизованих систем управління.

Фундаментом концепції побудови мереж наступного покоління можна виділити ідею створення такої мережі, яка б дозволяла користувачам обмінюватись між собою будь-якими видами інформації, до яких можна віднести аудіо, відео, текст, графіка, додатки та багато інших даних. В той же час ця система мала бути дуже гнучкою у питанні персоналізації та управління. Такого результату можна досягти за рахунок уніфікації мережних рішень. Також це допускає реалізацію такої мережі, яка може бути інтегрована з традиційними мережами зв'язку, та мати розподілену комутацію.

1. ЗАГАЛЬНИЙ ОГЛЯД СИСТЕМ УПРАВЛІННЯ

1.1 Поняття системи управління

Україна, як частина Європи та як країна, зі стрімким розвитком телекомунікаційних технологій, має перед собою задачі: надавати своїм користувачам весь спектр інформаційних послуг швидко та якісно. Фундаментом цього величезного комплексу по обміну, обробці та управлінню інформацією є телекомунікаційна мережа. Щоб залучити якомога більше споживачів та реалізувати вимоги ринку, тобто, зробити доступність глобальною, необхідно вирішити питання архітектури цієї мережі, тим самим оптимізувати підтримку нових систем застарілим обладнанням. Такі вимоги вимагають та сприяють створенню глибокої інформаційної інфраструктури, такої, що передбачена концепцією інтелектуальних мереж (ІМ). Об'єктивна оцінка стану мережі телекомунікацій наразі приводить до жорсткого висновку: перспективи розвитку базуються на тому, наскільки успішно та доцільно будуть використовуватися можливості існуючих та вже працюючих систем зв'язку. ІМ ґрунтується на основній ідеї розділення функцій надання послуг а також комутації, що за собою тягне можливість впровадити глобальну інформаційну інфраструктуру, що в свою чергу, дозволяє у будь де і у будь який час якісно та за відповідну плату надати користувачу будь-який вид телекомунікаційних послуг. Для забезпечення цих вимог потрібно продумати більш гнучку архітектуру системи управління, аніж ту, що є наразі. Вона має підтримувати швидку реалізацію та підтримку нововведень по всій глобальній мережі.

Системою управління називають набір систематизованих засобів впливу на об'єкт, що контролюється для досягнення поставленою йому задачі. Таким об'єктом може бути як машина так і людина. Об'єкт може складатись з безлічі інших об'єктів, що мають взаємозв'язки.

Об'єктом може бути якась динамічна система або модель цієї системи.

Охарактеризувати стан об'єкту можна змінними. В природних процесах цими змінними є температура, щільність речовин, курс (наприклад, цінних паперів).

Для машини такими змінними можуть бути швидкість переміщення, температури роботи, електричні змінні.

Такі системи управління можна розділити на два основні класи:

- Автоматизовані системи управління (АСУ) – людина приймає участь у управлінні такою системою;
- Системи автоматичного управління (САУ) – людина не приймає участь у управлінні такою системою.

1.2 Загальні поняття про мережу управління телекомунікаціями

Головною задачею управління телекомунікаціями є забезпечення якісної та тривалої роботи мереж та засобів комунікації під час їх невинного вдосконалення та розвитку в умовах змін. Телекомунікаційні мережі це об'єкти управління зі складною структурою, які потребують розподілену систему. Управління - це багатофункціональний процес.

Якщо брати загальний вигляд, що відповідає функціональному призначенню видів управління телекомунікаційними послугами та мережами можна згрупувати як функціональне, оперативне, координаційне та технічне (адміністративне управління і управління розробками і розвитком) управління. Реалізованими та взаємозв'язаними види управління можуть лише за умови існування системи управління, що є комплексною, тобто такою, що виконує функцію управління їх взаємодій.

Згідно з міжнародною моделлю система управління мережею зв'язку будується ієрархічно і має такі рівні (рис. 1.1): мережних елементів; управління елементами; управління власне мережею; управління обслуговуванням; адміністративного управління.

Найнижчий рівень являє собою саму мережу зв'язку, тобто об'єкт управління. Мережними елементами тут можуть розглядатися комутаційні станції, системи передавання, мультиплексори, комплекти обладнання для тестування тощо.

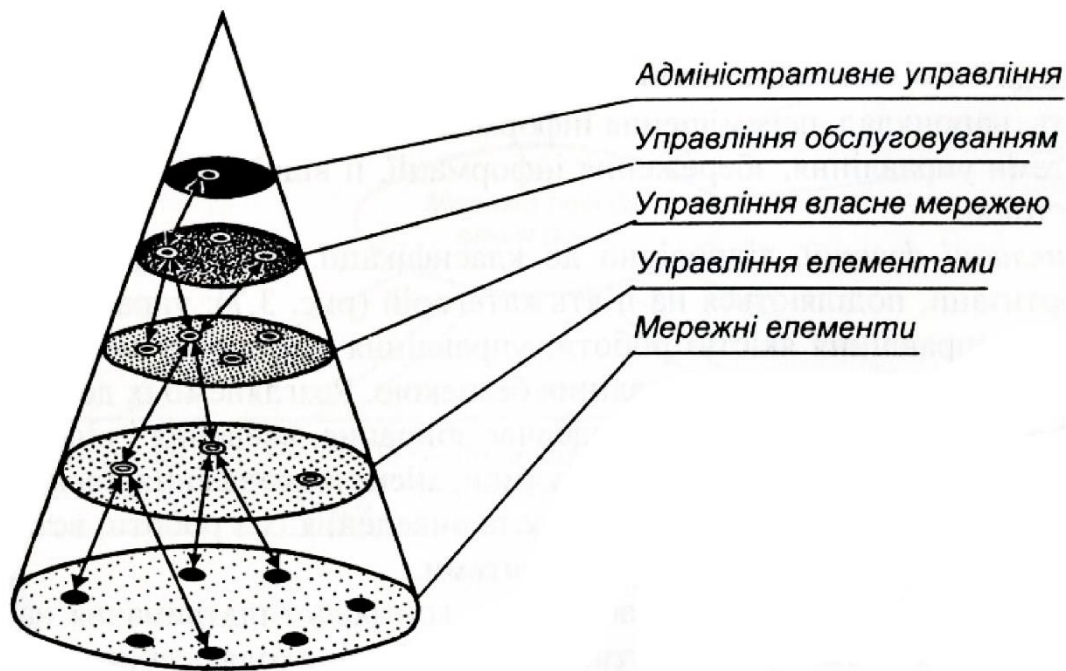


Рисунок 1.1 - Рівні управління мережею зв'язку

Кожен наступний рівень має вищий від попереднього рівня ступінь узагальнення. Дані про стан рухаються догори, в той час як керуючі впливи йдуть згори вниз. Процент автоматизації може бути будь-яким, проте частіше за все поєднують автоматизовані дії, та дії, що має зробити людина механічно. Так склалося, що чим вищий рівень управління, тем менше там автоматизації, більше автоматизму.

Управління елементами та підпорядкованими до них множинами, показ робочих параметрів, тестування та сервіс надається завдяки рівню управління елементами.

Контроль підпорядкованих до елементів множин, що мають взаємозв'язок між собою та іншими ресурсами мережі, можливість моніторингу системи вцілому надається завдяки рівню мережного управління.

Старт або зупинка послуг, облік, планування, якість обслуговування визначається на рівні управління обслуговуванням. Цей рівень відрізняється від нижчих тим, що має прямий зв'язок з мережею, та звернений до користувача.

Фінансові питання, питання організації, взаємодія з операторами, що представляють інші мережі зв'язку забезпечує рівень адміністративного управління. Тобто, можна сказати, що він забезпечує функціонування оператора мережі.

На сьогоднішній день наявні такі системи управління, що мають доступ до рівня управління елементами, а далі й управління мережею. У виключних випадках також і до управління обслуговуванням.

Можна виділити дві глобальні частини функцій, що пов'язані з управлінням: прикладні та загальні.

Передача інформації між елементами мережі, архівація, видача, сортування інформації та її пошук відноситься до загальних функцій. Також загальні функції підтримують прикладні..

У прикладних функцій можна виділити 5 категорій (рис. 1.2): конфігурація, усунення несправностей, безпека, розрахунки, якість роботи. Розглянемо їх докладніше.

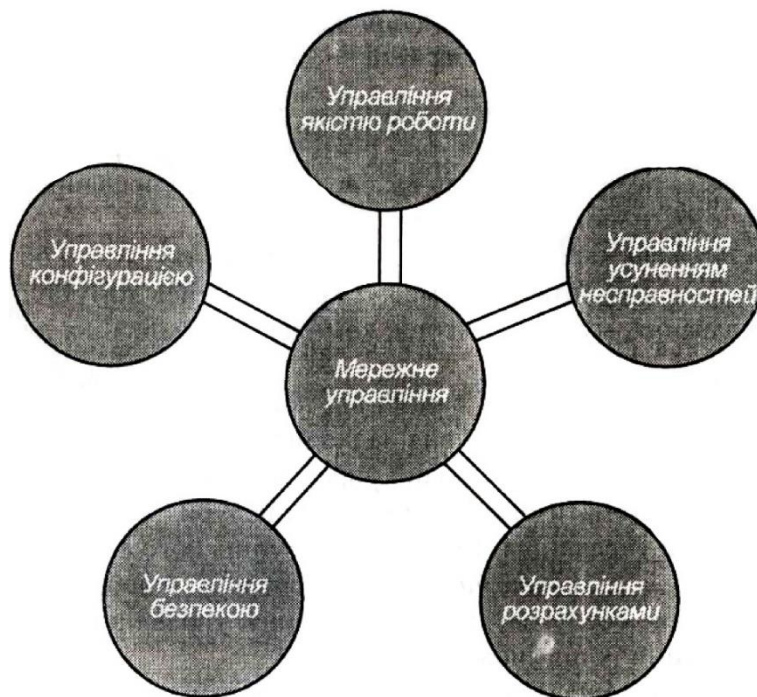


Рисунок 1.2 - Класифікація функцій мережного управління

Операції з мережними елементами, такі як їх збір, додавання, та відображення інформації про них (місцезнаходження, ID); початок та зупинка роботи елементів; налаштування та реконфігурація фізичних з'єднань між елементами забезпечує управління конфігурацією.

Збір, реєстрація, архівація, обробка, відображення інформації по мережі, тобто, моніторинг роботи мережі надається рівнем управління якістю. Цей рівень відповідає за підтримку та контроль головних характеристик мережі

Пошук, виокремлення, місцезнаходження несправності та її запис у реєстр виконує функція усунення несправностей. Також ця функція має можливість видавати користувачу список підказок/рекомендацій щодо усунення тієї чи іншої проблеми.

Все що стосується плати за використані мережні ресурси та їх контролю відноситься до функції управління розрахунками.

Захист за допомогою криптографії, виявлення несанкціонованого доступу виконується функцією управління безпекою. Ця функція може обмежити доступ,

якщо користувач не має відповідного паролю або активувати сигнал тривоги у разі, якщо доступ все ж таки було отримано несанкціоновано, а також розірвати таке несанкціоноване з'єднання.

Такий об'єкт, як телекомунікаційна мережа, можна назвати складним. Через це, вона потребує розмежованої системи.

Окрім цього слід зазначити що сучасні мережі набувають статусу інтелектуальних систем. Саме тому такі мережі потребують управління за протоколами, що не мають залежностей від виду надаваних послуг, однак вони можуть надаватися якими завгодно постачальниками, навіть такими, що є прямими конкурентами, по всій мережі та не бути залежними від технічних засобів, що задіяні у цій мережі. Ці умови можна бути виконати лише при використанні розподіленої системи управління.

Для того, щоб розподілені компоненти управління взаємодіяли в одній системі, а також для виконання ними функцій управління необхідна мережа, що передає інформацію управління. Така задача утворила потребу у створенні концепцію управління телекомунікаціями TMN (Telecommunication Management Network) . Мережа, що побудовано на концепції управління передбачає виконання управління послугами комунікацій та мережами вцілому, також надаючи зв'язок між внутрішніми частинами, нею самою і телекомунікаційними мережами, послугами та іншими мережами управління телекомунікаціями (іншими TMN).

Здійснення розподілу функцій мережі таким чином, що з'являється можливість реалізації централізованого або навпаки управління відбувається завдяки принципам логічного відділення МУТ від мереж та послуг, які під її керуванням. Тобто мережа управління телекомунікаціями дає змогу операторам телекомунікацій з кількох центрів здійснювати управління широкою номенклатурою розподіленого обладнання, мереж та послуг.

Відповідно до рекомендацій ITU мережа управління телекомунікаціями призначена для виконання завдань управління мережами телекомунікацій та службами операторів телекомунікацій при плануванні мереж, наданні послуг,

побудові та введенні в експлуатацію нових систем і послуг, технічному обслуговуванні й адміністративному управлінні. При цьому повні функціональні можливості TMN полягають у спроможності:

- обміну інформацією управління через стики TMN з мережами телекомунікацій;
- обміну інформацією управління через стики між нею самою й іншими мережами управління телекомунікаціями (іншими TMN);
- перетворення інформації управління з одного формату в інший таким чином, щоб інформація, що циркулює в її межах, мала однакову природу;
- передавання інформації управління між пунктами в межах TMN;
- аналізу й адекватного реагування на інформацію управління;
- обробки інформації управління і доведення її до форми, зрозумілої користувачеві цієї інформацією;
- доставки інформації управління користувачеві цієї інформацією і відображення її в належному вигляді;
- гарантія обмеження доступу до управління несанкціонованим користувачем.

Будь-яку систему є змога розділити на дві фундаментальні частини: прийняття рішень та виконання рішень. Прийняття відбувається під виглядом ОС, виконання відбувається за допомогою апаратної складової. Якщо взяти цю інформацію до уваги, то виходить, що принцип TMN заключається в тому, що надається спеціальна організаційна архітектура, що дозволяє та гарантує взаємозв'язок між програмною складовою (у нашому випадку це різноманіття ОС) та апаратною складовою (телекомунікаційне обладнання), для того, щоб вони могли обмінюватись інформацією, використовуючи архітектуру, що використовує стандартні інтерфейси, протоколи та повідомлення.

Концептом TMN є окрема мережа, яка під'єднується до основної мережі телекомунікацій у відповідних вузлах і дозволяє за допомогою обміну інформацією управляти цією основною мережею. Для підтримки роботи TMN

може, подібно симбіозу, використовувати частини основної мережі. Також TMN використовує системи комутації основної мережі для підтримки своєї роботи.

Використовуючи TMN, для приведення прикладу, можна керувати такими складовими, як:

- мережі загального користування. Сюди також можна віднести цифрові мережі з інтегральним обслуговуванням, також мережі, які зв'язані з об'єктами, що рухаються, а також з іншими віртуальними та інтелектуальними мережами;

- управління телекомунікаційною мережею;
- обладнання систем передачі;
- системи передачі (аналогові та цифрові, тобто, як кабельні так і

супутникові)

- системи резервного копіювання;
- ОС та периферія (процесори, сервери, контролери);
- WAN та LAN мережі.;
- мережі в яких канали з'єднуються з пакетами;
- обладнання та системи сигналізації, бази даних;
- телематичні служби;
- АТС, їх точки доступу та термінали;
- термінали користувачів цифрових мереж (ISDN);
- ПЗ телекомунікаційних служб та їх похідних, для прикладу, комутаційне

ПЗ;

- прикладне ПЗ для процесора, включаючи ПЗ для TMN;
- системи контролю безпеки та якості, такі як кондиціонування,

електроживлення, сигналізація (також пожежна), обладнання для тестувань.

Перелічені системи TMN існують для того, щоб гарантувати відповідний рівень роботи починаючи від введення в експлуатацію та при умові постійного вдосконалення.

Основними завданнями TMN є:

- під час введення мереж в експлуатацію - планування мереж та створення баз даних;

- у процесі експлуатації мереж - технічне обслуговування, відновлення зв'язку при пошкодженнях, управління трафіком, контроль за якістю послуг, розрахунки із користувачами;

- у процесі розвитку мереж - прогнозування зростання трафіка, модернізація мереж.

ITU рекомендує розділяти системи управління на функціональні області, що визначаються за призначенням, задля більш ефективного та раціонального управління мережею управління. З цього маємо п'ять областей:

- управління робочими показниками;
- усунення несправностей;
- конфігурація;
- безпека інформації;
- управління розрахунками (Accounting Management).

На кожному логічному рівні системи управління телекомунікаціями вирішується свій перелік завдань,. При меншій, ніж наведено в таблиці, кількості рівнів управління, коли два або більше рівнів об'єднуються в один, на ньому вирішуються завдання, притаманні кожному із рівнів управління, що входять до об'єднання.

Слід зазначити, що наведені функції управління:

- не є обов'язковими для будь-якого елемента мережі або мережі управління телекомунікаціями. Їхня специфікація і розвиток є внутрішньою справою адміністрацій зв'язку;

- не вичерпують увесь перелік функціональних областей управління. Наприклад, як окремі області управління розглядаються також введення в експлуатацію, планування, прогнозування і деякі інші.

Взаємозв'язок між TMN та мережею зв'язку показаний на рис.1.3. Операційні системи здійснюють оброблення інформації, необхідної для виконання функцій

управління. Робочі станції забезпечують користувальницький інтерфейс, за допомогою якого обслуговуючий персонал взаємодіє з мережею управління. Мережа передавання даних призначена для зв'язку між мережними елементами, операційними системами та іншими компонентами TMN.

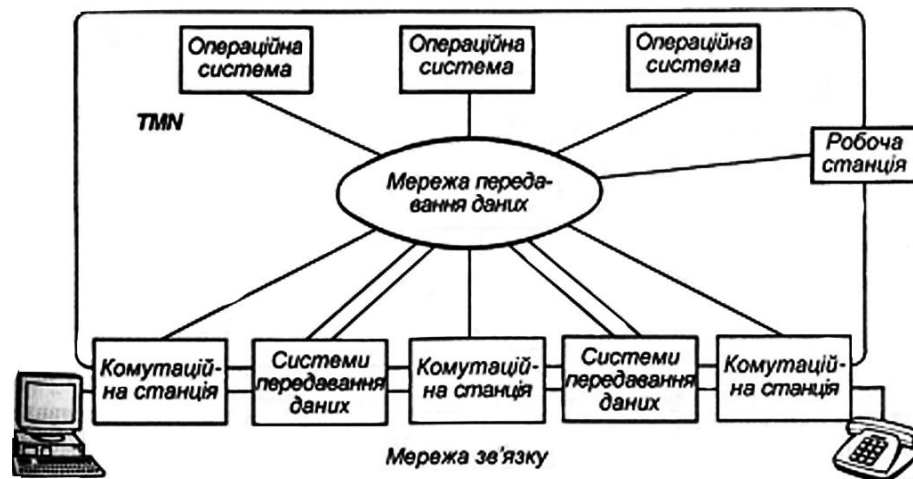


Рисунок 1.3 - Взаємозв'язок між мережею управління та мережею зв'язку

Спрощена фізична архітектура мережі управління (рис.1.4), крім згаданих компонентів, має у своєму складі проміжні пристрої (медіатори), інтерфейси, призначені для взаємодії між компонентами, та Q-адаптери для взаємодії з мережними елементами або операційними системами, які мають непередбачені в TMN інтерфейси.

Медіатори використовуються для проміжного оброблення та збереження даних і перетворення протоколів. Вони не є необхідним компонентом, тому що їхні функції можуть виконуватися безпосередньо в мережних елементах.

Інтерфейс F служить для зв'язку робочих станцій з іншими компонентами.

Інтерфейс X призначений для взаємодії з іншими мережами, яка здійснюється через мережу передавання даних.

Інтерфейси Q забезпечують взаємодію мережних елементів, операційних систем і медіаторів через мережу передавання даних. Інтерфейс Q₃, який відіграє

центральну роль у TMN, служить для стикування мережі передавання даних з операційними системами, медіаторами та мережними елементами, які виконують функції медіатора. Інтерфейс Q_x використовується для приєднання мережних елементів до медіатора.

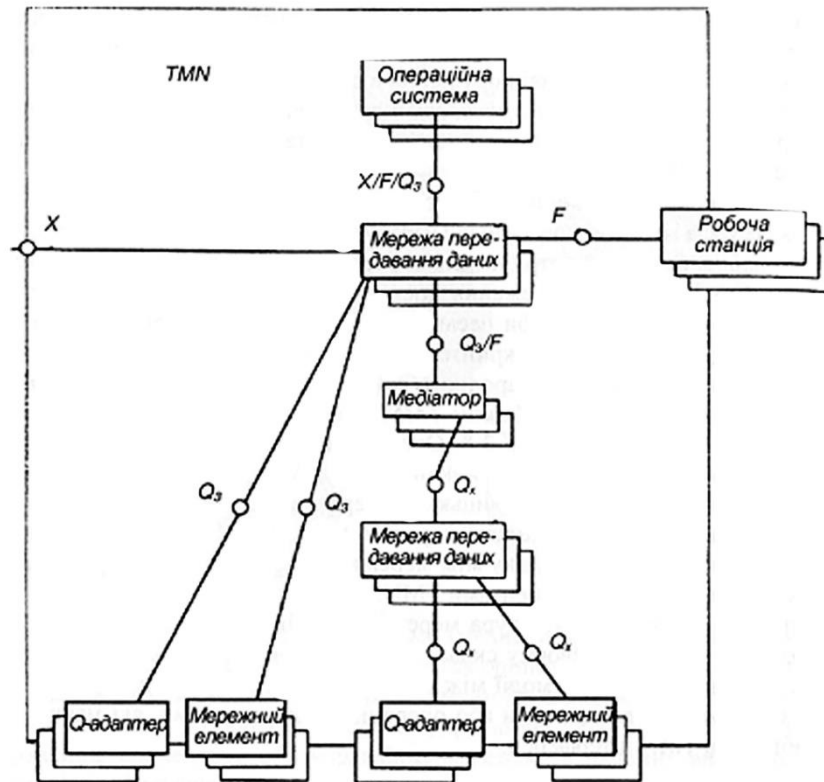


Рисунок 1.4 - Спрощена фізична архітектура мережі управління

1.3 Стандарти систем управління

У світі існує багато стандартів систем управління, проте популярними можна виділити дві: це стандарти Internet, що керують на основі протоколу SNMP (Simple Network Management Protocol) та CMIP (Common Management Information Protocol) на які спираються міжнародні стандарти OSI, розроблені ISO і ITU-T. Перша серія стандартів мінімально специфікує аспекти і елементи управління, в той час як друга серія навпаки — максимально. У сучасному світі так склалося, що корпоративні та локальні мережі для себе використовують SNMP, а ISO/ITU та

СМІР використовують якраз таки для управління мережами та послугами телекомунікацій.

SNMP базується на концепції мінімального навантаження:

- агент виконує лише прості функції і виконує їх лише за викликом;
- у системи мається єдиний менеджер, який взаємодіє та опитує усіх агентів;
- взаємодія агента та менеджера спирається на простий UDP, що дозволяє

зняти навантаження з керованого пристрою, та використовує лише дві основні команди – get для опитування та set для передачі даних;

- у деякому проценті випадків, агент може використати незвичну для нього команду trap для відправки даних, проте ймовірність цього незначна;

MIB у стандартах Internet можна розкласти на дерево атрибутів. Це об'єкти та групи об'єктів.

Перші MIB Internet керувалися за допомогою маршрутизаторів: MIB-I - тільки контроль, MIB-II - контроль і управління. У подальшому була розроблена RMON MIB. Вона була націлена на те, щоб створювати розвинутих агентів, які б могли контролювати нижчий рівень, тобто Ethernet та Token Ring. У стандартних MIB Internet імена об'єктів зареєстровані ISO.

Стандарти ISO/ITU-T для віртуалізації керуємих пристроїв для уявлення пристроїв, якими управляють, використовують об'єктно-орієнтований підхід. Для узагальнених керованих об'єктів визначено декілька суперкласів. Базуючись на них та успадковуючи їх властивості мають бути створені більш направлені класи.

Саме для того, щоб описати OSI було розроблено відповідні правила GDMO, що базуються на виокремленій структурі, яка заповнюється мовою ASN.

OSI використовує деревоподібні БД для опису агентів, менеджерів та об'єкти управління: спадкування між класами об'єктів; включень – описує супідрядні відносини між елементами; імен – визначає імена у системі.

Взаємозв'язок агентів і менеджерів OSI визначається протоколом СМІР. Він дозволяє одним запитом задіяти множину агентів та надати їм відповідно необхідну задачу, наприклад, фільтрацію або огляд.

1.4 Реалізація систем управління мережами та послугами телекомунікацій

Слід зазначити, що ще в колишньому СРСР у планах створення Єдиної автоматизованої системи зв'язку (ЄАСЗ) вирішенню питань розробки і впровадження автоматизованих систем управління (АСУ) мережами телекомунікацій приділялася велика увага.

При створенні сучасної автоматизованої системи управління мережами телекомунікацій можливі кілька сценаріїв (рис. 1.5)

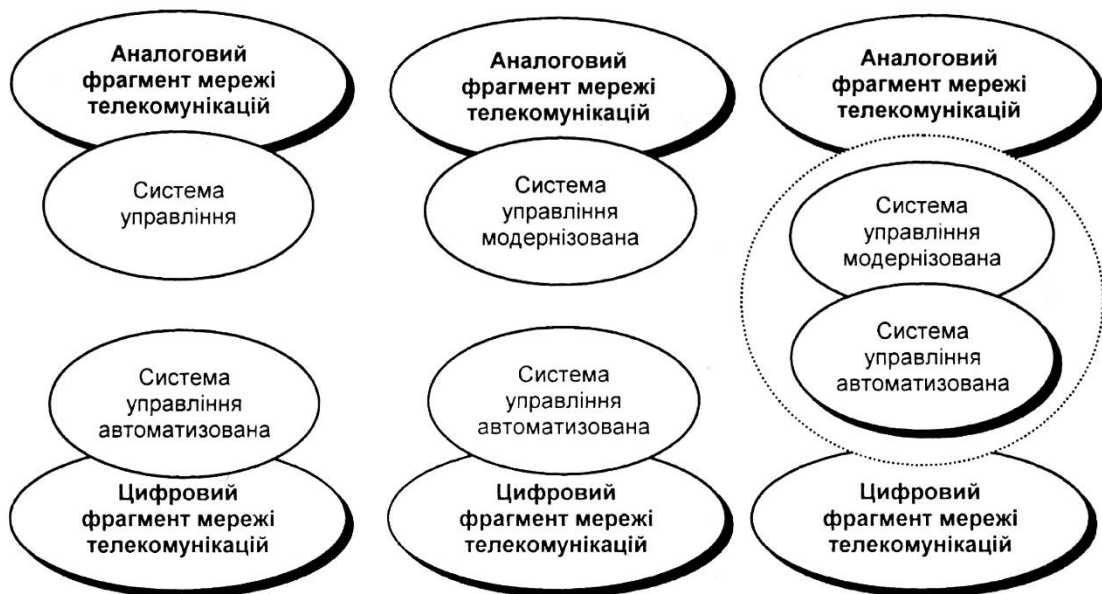


Рисунок 1.5 - Сценарії створення автоматизованої системи управління

За першим сценарієм створюється окрема автоматизована система управління цифровим фрагментом мережі телекомунікацій. Існуюча система управління аналоговим фрагментом мережі телекомунікацій не модернізується, а функціонує за старими алгоритмами і технічними засобами.

За другим сценарієм, крім створення окремої автоматизованої системи управління цифровим фрагментом мережі телекомунікацій, модернізується існуюча система управління аналоговим фрагментом. Кожна з цих систем використовує свої окремі обладнання, програмне забезпечення і базу даних.

За третім сценарієм в кілька етапів створюється автоматизована система управління цифровим фрагментом мережі телекомунікацій. Система управління аналоговим фрагментом мережі поступово частково зливається з системою управління цифровим фрагментом (використовуючи спільну базу даних, спільну операційну систему, сучасні технічні засоби) з автоматизацією тих процесів, які піддаються автоматизації з найменшими витратами.

Найбільш прийнятним вбачається третій сценарій, оскільки він дає можливість об'єднати управління технологічно різними фрагментами мереж, які ще певний час мають співіснувати. Повна автоматизація процесів управління застарілим аналоговим обладнанням, як уже зазначалося, пов'язана з суттєвими капітальними витратами, які перевищують експлуатаційні витрати за весь період залишкової експлуатації цього обладнання, що підтверджується досвідом провідних західних операторів. Крім того, значна частина функцій управління застарілим обладнанням принципово не може бути ефективно автоматизованою.

На рис.1.6 зображено служби, з яких може складатися гіпотетичний центр управління телекомунікаціями.

Кожна із служб центру управління несе відповідальність за управління певним конкретним видом мереж (транспортною, телефонною, документального зв'язку та ін.), за загальне управління мережами (управління трафіком, синхронізацією тощо), за виконання інших функцій системи управління (управління технічним обслуговуванням, управління системою безпеки, розрахунками, взаємодією з іншими операторами, якістю, персоналом та ін.).

Взаємодія між службами забезпечується на рівні операційних систем кожної із служб (підсистем системи управління) через єдину мережу передавання даних.

Наведена структура є лише прикладом. Практично в повному обсязі вона може бути реалізованою на рівні НЦУ, ГЦУ. На інших рівнях управління створюються служби (підсистеми системи управління), необхідні на даному рівні, або у даного оператора. Можливі групування або інші комбінації служб.

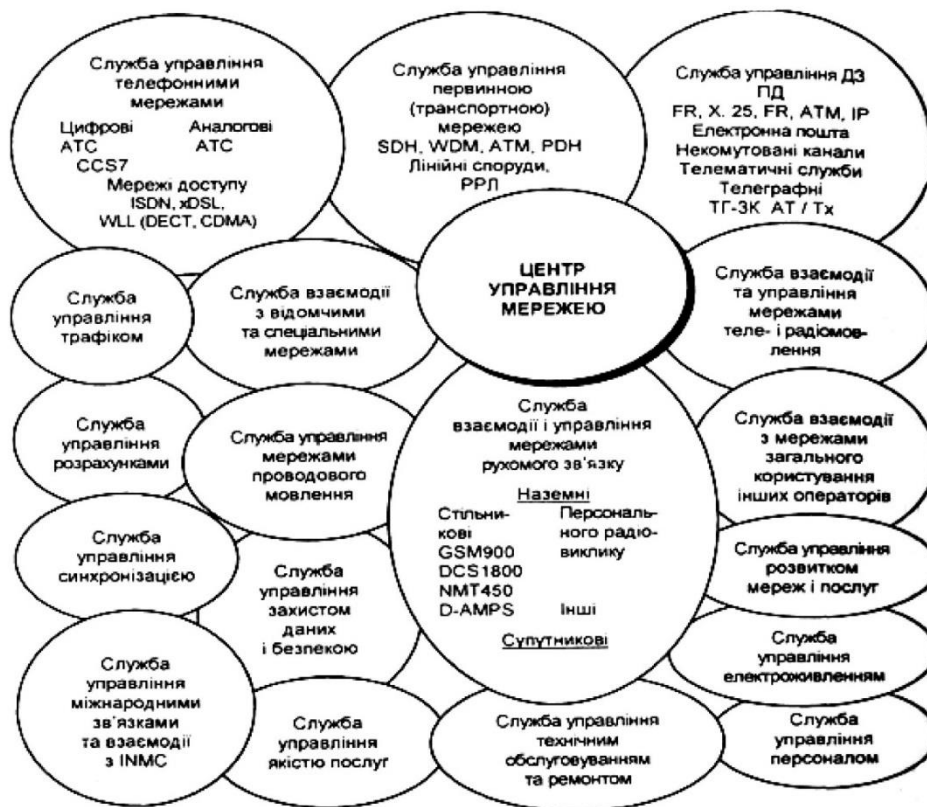


Рисунок 1.6 - Структура гіпотетичного центру управління мережами телекомунікацій

1.4.1 Система управління мережами NGN

Фундаментом концепції побудови мереж наступного покоління можна виділити ідею створення такої мережі, яка б дозволяла користувачам обмінюватись між собою будь-якими видами інформації, до яких можна віднести аудіо, відео, текст, графіка, додатки та багато інших даних. В той же час ця система мала бути дуже гнучкою у питанні персоналізації та управління. Такого результату можна досягти за рахунок уніфікації мережних рішень. Також це допускає реалізацію такої мережі, яка може бути інтегрована з традиційними мережами зв'язку, та мати розподілену комутацію.

Концепція NGN заключається в розподілені комутації, переносу, управління кикликами та послугами.

Функціональна модель мереж NGN представлена на рис. 1.7.

Загально можна представити таку модель трьома рівнями:

- транспортний;
- управління комутацією і передачею інформації;
- управління послугами.

Транспортний рівень вирішує питання комутації та передачі даних до споживача.

Управляє потоками, маршрутизує виклики та обробляє сигналізацію рівень управління комутацією та передачею.

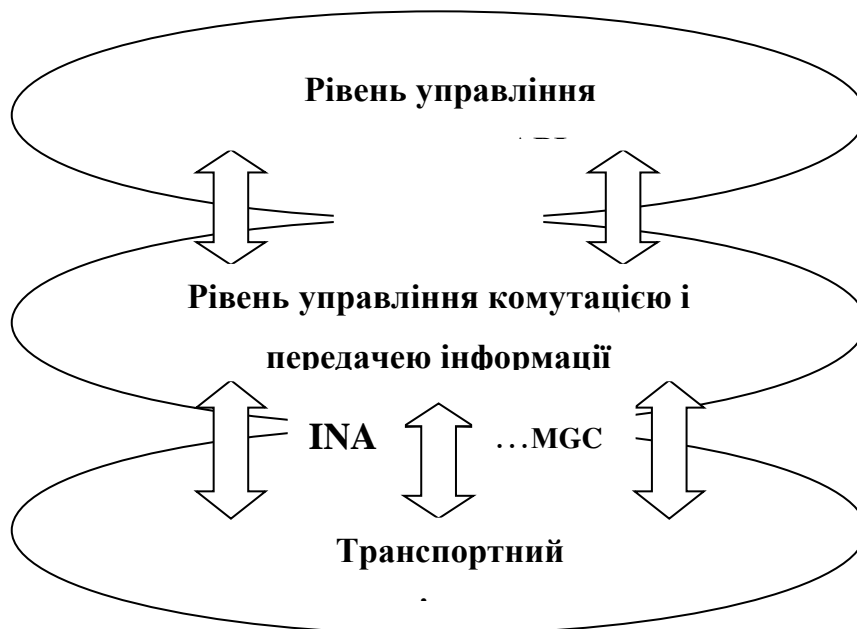


Рисунок 1.7 – Функціональна модель мереж NGN

Логікою послуг і додатків займається рівень управління послугами.

Насамперед, цей рівень забезпечує:

- взаємодію різноманітних послуг;
- управління цими послугами;
- створення та впровадження нових послуг до вже існуючих;
- можливість використання комунікаційних послуг.

На такому рівні є можливість використати програму логіки доступу, яка не буде залежати від способу цього доступу або від типу транспортної мережі. Саме існування цього рівня дозволяє нам створювати та запроваджувати нові послуги таким чином, що ми не будемо втручатися у роботу інших рівнів.

Цей рівень може включати у собі безліч підсистем, що незалежні одна від одної, а також працюють на основні різних реалізацій, мають різних абонентів, та адресують їх своїми внутрішніми системами.

Дефолтні системи управління являють собою множину агентів, які займаються лише тим, що збирають статистику з пристроїв мережі та виконують найпростішу обробку для того, щоб потім відправити ці дані на великий сервер, що займається управлінням. При побудові новітніх систем управління у наш час, було прийнято рішення відмовитись від такої ієрархії у бік федеративного принципу управління.

Агент мереж різноманітних корпорацій не стоять на місці та з кожним роком все більш удосконалюються. Приєднані до відповідних вузлів або пристроїв вони мають змогу керувати ними, базуючись на інформації, яку вони збирають навколо себе, а також керувати ресурсами, підконтрольними тому вузлу чи пристрою. У сучасних системах моніторинг та аналіз не потребує передачі даних кудись у спеціальне місце, а виконується локально, агентом. Центру обробки інформації тепер не потрібно навантажувати себе безліччю інформаційного сміття, яке потрібно обробляти, та затратити на це значні потужності, за них це зробить агент, а сам центр тепер виконує роль «диригента», який лше дає команди та приймає рішення. Таким чином ресурси використовується якомога раціональніше. Також можна виділити, що при такому підході відпадає проблема масштабування, бо збільшення кількості агентів навантажує систему не пропорційально.

Інтелектуальні агенти дозволяють повною мірою реалізувати принцип "проактивності". Потенційна проблема на мережі може бути ідентифікована інтелектуальним агентом по характерним симптомах і ліквідована за допомогою

превентивних мір. Агент повинен довести до відома сервера інформацію про виконані дії, замість того щоб очікувати реакції від оператора на інформаційне повідомлення.

Незважаючи на "федеративний" перерозподіл, значення сервера управління не тільки не знижується, а, навпаки, зростає. Саме сервер реалізує адаптивність мережі, координуючи і направляючи роботу агентів для досягнення поставлених цілей, підключаючи додаткові ресурси якщо виникає потреба.

До характерних рис NGN, з боку управління можна віднести те, що вони складаються з величезної множини компонентів різного призначення. Слід зазначити, що раніше такі системи складались з порівняно невеликої множини комутаційних пристроїв. Окрім цього можна додати, що у новій системі буде організована підтримка більшого числа інтерфейсів, порівняно із сьогоденними системами. Оперуючи цими фактами можна зробити висновок, що є необхідність перегляду підходів і принципів використання NGN.

Така система управління має включати в себе встановлений об'єм рішень, який зможе надати адекватне управління мережами, які, в свою чергу базуються на різних технологіях та напрямленнях (наприклад, стільникова, сигналізація, передача інформації) і використовую обладнання та технічну базу великої кількості різних виробників. Ця система управління буде реалізовуватись за допомогою ОО та розподіленої структури.

NGN розроблена таким чином, що надає змогу використовувати вже розроблені та працюючі модулі, а також модернізувати їх для подальшого використання, тобто, можна сказати, що ця система має модульну структуру. Для того, що мати можливість користуватись системами й мережами, які розроблені різними виробниками використовуються загальні протоколи, такі як, наприклад, SNMP, OSI, ASCII. Можна зазначити, що дефолтним інструментом в таких мережах являється SNMP. Також, для TMN дефолтною був призначений протокол OSI. Як згодом виявилось на практиці, така система була складна в реалізації, але навіть при умові її реалізація вона була недопустимо повільною та

потребувала нерентабельний рівень ресурсів для підтримки роботи. Окрім цього вона не задовольняла потрібний рівень ефективності у питанні управління. Наразі ведуться дослідження та продумується реалізація на основні архітектуру CORBA. Це обумовлено перспективністю такої архітектури, насамперед, на верхніх рівнях управління.

Одна із задач поставлених перед СУ – відкритий інтерфейс. Серед інтерфейсів, які мають таку рису можна відмітити POP, CMIP, SNMP, FTP; серед мов - IDL, JAVA, ASN; такий набір гарантує стабільну роботи при внесенні змін, бо внесуться лише ті зміни, що взаємосумісні, тобто, не буде аварійних зупинок роботи.

Для наведення прикладу можна зазначити, що аварійні повідомлення будуть посилатися за допомогою CMIP, SNMP або CORBA та будуть використовувати ОМ, що описана в X.733; для організації послуг можуть використовуватися інтерфейси CORBA; для пересилання даних про робочі характеристики може застосовуватися протокол FTP.

Основними задачами перед NGN є:

- змодельоване рішення має будуватися на практиці в мінімальний термін;
- архітектура відкритої системи має гарантувати легкість сумісності та взаємодії з паралельними реалізаціями. Також треба наголосити на гарантії надійності та витікаючи з цього – якості обслуговування;
- користувачу має бути надана можливість модифікації ПЗ для впровадження нових функцій та послуг;
- підтримка модульності. Модульність полегшить роботу оператора в напряму додання нових користувачів та функціоналу.

Адаптація нових технологій підтримується завдяки тому, що система реалізована гнучкою та з можливістю масштабування.

З економічної точки зору впровадження такої системи раціональне через низький рівень витрат ресурсів на управління та технічну підтримку, інфраструктуру, а з іншого боку, збільшення ефективності інформаційних ресурсів загалом.

1.4.2 Система управління мережами SDH

СУ мережею (цифрової магістралі), що реалізована на базі технології та входить до СУ загальної мережі, та складається з полуавтоматичних СУ окремими мережами. Працюють гуртом ці системи за допомогою рівня центрів управління (рис. 1.8).

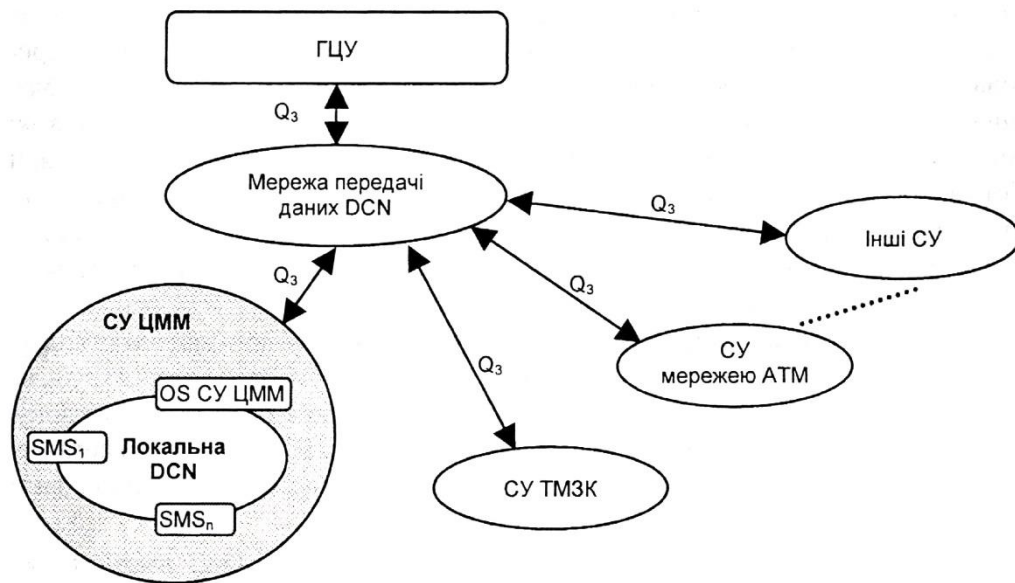


Рисунок 1.8 - Місце СУ ЦММ (SDH) у загальній структурі системи управління мережами телекомунікації

Контроль систем передачі, що являють собою частини магістральної мережі SDH, виконується впровадженими підсистемами SMS_j , \dots , SMS_m .

SMS гарантує контроль частин мереж, що реалізовані або тільки реалізуються використовуючи технічні рішення різноманітних постачальників (Ericsson, Nortel, Lucent Technologies), що працюють за допомогою множини протоколів, моделей, інтерфейсів.

1.4.3 Система управління телефонними мережами

СУ загальними телефонними мережами яка росте за допомогою впровадження у мережу новітніх цифрових технологій, є частиною СУ мережей, яка в свою чергу складається з напівавтоматичних СУ окремими мережами. Взаємозв'язок та нормальну роботу цих систем підтримує рівень центру управління (рис.1.9).

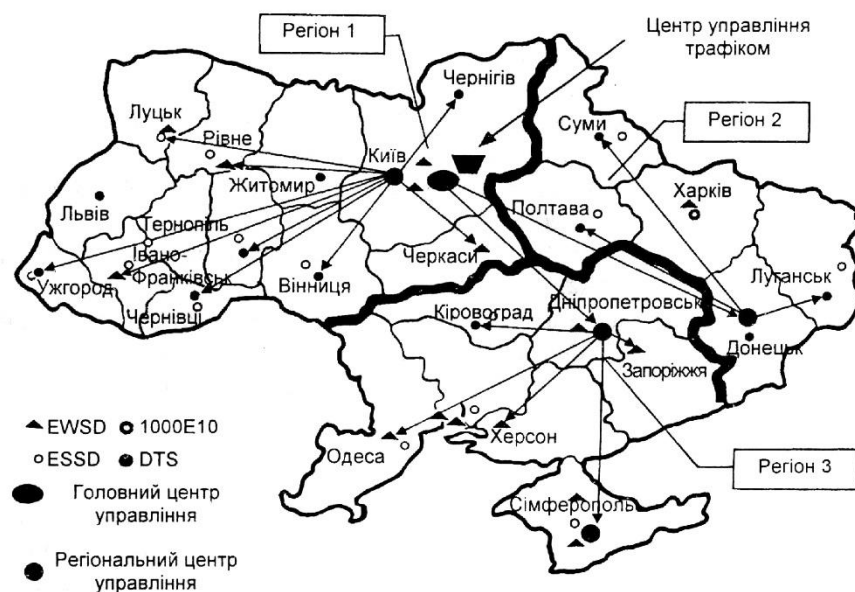


Рисунок 1.9 - Загальний вигляд розміщення СУ ТМЗК

В цілому СУ ТМЗК має виконувати такі основні:

- централізованого контролю працездатності мережі, автоматичного виявлення та діагностування пошкоджень;
- централізованого контролю та дистанційного управління конфігурацією обладнання і мережі в цілому;
- збирання та аналізу даних щодо якісних показників функціонування обладнання, мережі та послуг, інформаційної підтримки прийняття рішень щодо вдосконалення якості роботи та послуг, динамічного управління трафіком;
- інформаційної підтримки планування, проектування та формування мережі;

- управління системою сигналізації CCS7 (або взаємодії з такою службою управління у разі її самостійного існування);

- взаємодії з головним центром управління певного оператора, з центром управління трафіком, з національним центром управління мережами зв'язку, з центрами управління інших операторів.

Базовою системою управління може стати існуюча система управління міжміською телефонною мережею UTEL, нині охоплює 23 АМТС, розташовані в усіх обласних центрах країни (рис.1.10). Вона складається з двох паралельно працюючих підсистем управління АМТС: MFOS - центр технічної експлуатації для комутаційних систем типу 5ESS та O&M NodeCommander - центр технічної експлуатації для комутаційних систем типу EWSD. Взаємодія систем MFOS та O&M між собою та верхнім рівнем управління здійснюється через операторів підсистем. Як мережа передавання даних використовується спеціальна модемна мережа. Об'єктами управління є цифрові АТС усіх рівнів ієрархії телефонної мережі UTEL - УКРТЕЛЕКОМ.

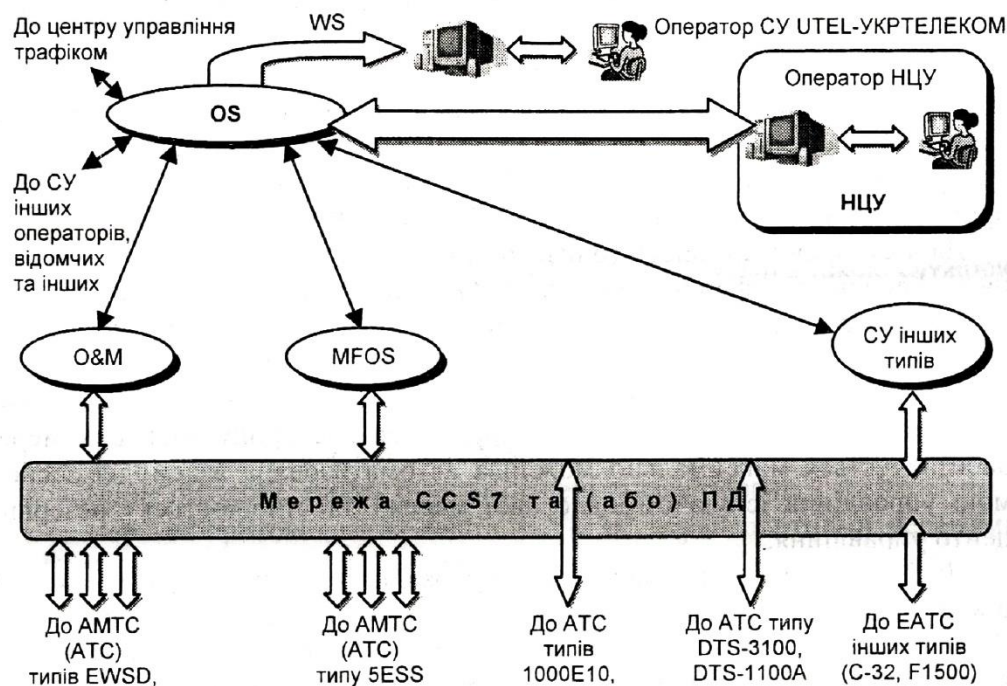


Рисунок 1.10 - Приклад функціональної структури СУ ТМЗК

Система управління телефонною мережею загального користування (СУ ТМЗК) призначена для забезпечення передавання трафіка користувача з нормованими надійністю і якістю, в необхідний термін кращого використання ресурсів мереж зв'язку, зниження експлуатаційних витрат, забезпечення надійності та живучості мереж шляхом здійснення якісного контролю мереж у реальному часі та оперативного управління мережами на всіх рівнях.

Дворівнева ієрархічна структура СУ ТМЗК має відповідати архітектурі TMN (M.3010).

На першому рівні розміщується центр управління системою (ЦУС), який повинен виконувати такі функції:

- взаємодіяти з головним центром управління домінуючого оператора та національним центром управління мережами телекомунікацій;
- динамічно управляти трафіком;
- управляти системою сигналізації CCS7;
- взаємодіяти з другим рівнем системи управління.

Другий рівень системи управління є рівнем регіональних центрів управління домінуючого оператора (приблизно 3-4) та центрів управління інших операторів.

Взаємодія між рівнями СУ ТМЗК має здійснюватися через мережу CCS7 або мережу передачі даних (МПД).

СУ ТМЗК має забезпечувати:

- управління конфігурацією мережі при організації зв'язків і наданні послуг користувачам;
- управління конфігурацією мережі для відновлення пошкоджених зв'язків;
- управління конфігурацією мережі з метою перерозподілу навантаження за результатами контролю трафіка;
- управління якістю роботи цифрової телефонної мережі загального користування шляхом безупинного контролю якості роботи комутаційних систем, управління трафіком, спостереження за якістю обслуговування.

СУ ТМЗК має охоплювати усі типи цифрових станцій телефонних мереж принаймні домінуючого оператора та здійснювати прийом інформації від станцій про стан навантаження і якість роботи.

СУ ТМЗК повинна опрацьовувати зібрані дані та обчислювати параметри управління мережею.

СУ ТМЗК має забезпечувати можливість визначення аварійних ситуацій на станціях.

СУ ТМЗК має контролювати процес ліквідації аварій і забезпечувати передавання станціям інформації про стан мережі та команд управління.

Повинна передбачатись можливість передавання інформації щодо управління мережею у реальному часі, за запитом, за розкладом та у випадках надзвичайних подій.

СУ ТМЗК має забезпечувати можливість активації/деактивації дій щодо управління відповідно до рішень, які прийняті.

Для прийняття рішення щодо управління мережею має бути передбачена наявність кількох порогових значень контрольних параметрів, які змінюються.

Повинна передбачатись можливість автоматичного, напіваавтоматичного та ручного режимів управління.

В СУ ТМЗК має забезпечуватись індикація на станціях та в центрах управління інформацією щодо управління за допомогою візуальних індикаторів та/ або друкуючих пристроїв або відеотерміналів.

СУ ТМЗК має забезпечувати планування (аналіз) ситуацій надзвичайного зростання трафіка в мережі.

В СУ ТМЗК потрібно передбачати можливість автоматичного документування інформації, прийнятої від елементів управління, даних про обчислені параметри, прийняті рішення, статистичних даних тощо.

1.4.4 Система управління трафіком

Виникнення перевантажень у мережі може бути непередбаченим і відбувається без якогось попередження. На умови експлуатації мережі негативно впливають такі фактори, як ушкодження систем передачі, комутаційних систем, заплановані відключення систем передачі і комутації, надзвичайне зростання потреби в обміні інформацією у користувачів, сконцентровані перевантаження і масові виклики, проблеми щодо можливості надання послуг і задоволення потреб користувачів внаслідок затримок у наданні додаткових каналів або іншого обладнання, перевантаження взаємозалежних мереж і т. ін.

Ситуації, що викликають різке зростання потреби в обміні, бувають передбаченими (свята, спортивні змагання, телеголосування тощо) або непередбаченими (стихійні лиха, катастрофи та ін.).

Зазначені фактори можуть бути причиною перевантажень, що негативно позначаються на якості обслуговування. Роботу мережі загального користування можна значно поліпшити, якщо заходи для усунення негативних наслідків такого роду явищ виконуватимуться оперативно.

Отже, виникає потреба створити систему оперативного або динамічного управління трафіком (рисунк 1.11), метою якої є функціональний захист мережі від будь-яких ситуацій, що негативно впливають на її роботу, зберігання високої пропускної спроможності з забезпеченням заданої якості обслуговування користувачів у звичайних умовах і в умовах надзвичайних ситуацій. Цього можна досягти шляхом прогнозування пікових ситуацій, моніторингу стану мережі і прийняття заходів для забезпечення працездатності мережі, впливу на трафік на будь-якій ділянці мережі (можливо її реконфігурування) для усунення проблем, що виникають. При цьому використовують наявні засоби систем передачі і комутації, що можуть бути оперативно перерозподілені в зв'язку з коливаннями навантаження або в аварійних ситуаціях.

Управління трафіком має поширюватися також на зменшення навантаження комутаційних систем і недопущення його збільшення. Збільшення кількості незадоволених викликів і, як наслідок, зростання кількості повторних викликів призводить до перевантаження комутаційної системи. Якщо з цим перевантаженням не боротися, воно може поширитися на сусідні комутаційні системи і призвести до ще більшого погіршення якості роботи мережі. Система управління трафіком має усувати повторні виклики в перевантаженій комутаційній системі (анулюючи, наприклад, трафік, спрямований в обхід через перевантажену комутаційну систему).

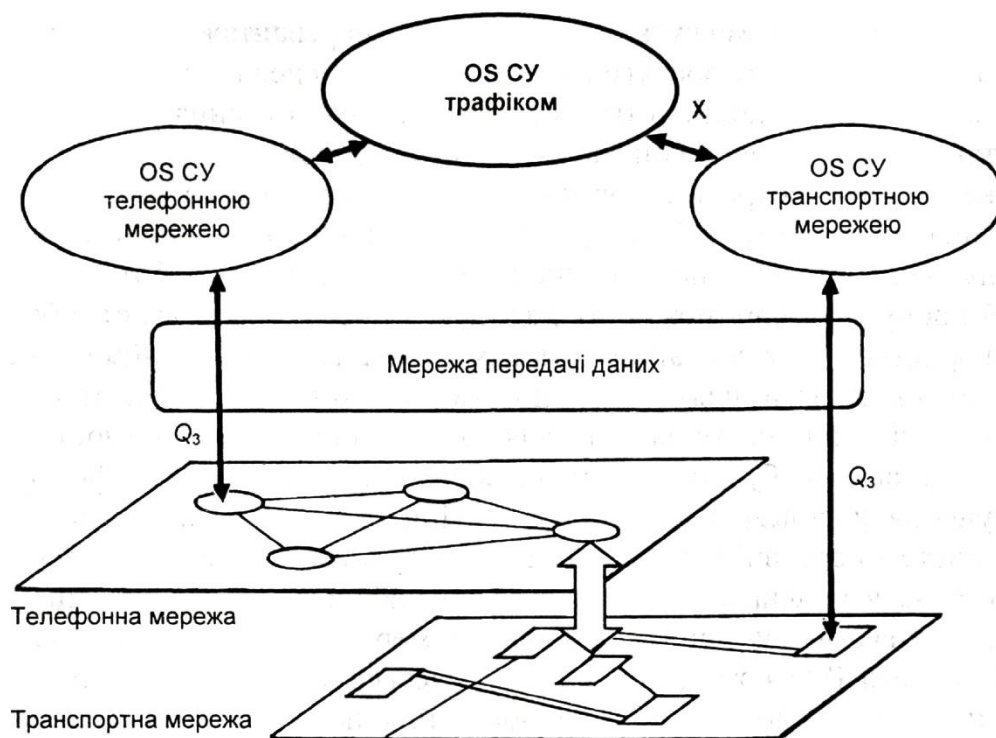


Рисунок 1.11 - Приклад організації управління телефонним трафіком

Реалізація системи управління трафіком в інтегрованій системі управління має такі переваги:

- збільшення прибутків внаслідок зростання кількості ефективних викликів;

- поліпшення якості послуг, наданих користувачам, та збільшення їх кількості за рахунок поліпшення з'єднань між користувачами, підвищення можливості проходження успішних викликів і, як наслідок, більшого визнання оператора телекомунікацій користувачами;

- ефективніше використання мережі і, як наслідок, збільшення віддачі капіталовкладень у мережу, поліпшення співвідношення ефективних викликів до неефективних;

- краще знання реального стану роботи мережі, що дозволяє встановлювати пріоритети в управлінні і технічному обслуговуванні мережі, поліпшити показники планування мережі та стосунки з користувачами;

- забезпечення в будь-який час захисту основних видів послуг і доступу до екстрених служб в екстремальних ситуаціях.

Отже, система управління - це складна динамічна система, до складу якої входять: цілі, функції, структура, методи управління, інформація, технічні засоби. Ці елементи тісно пов'язані між собою і взаємодіють один з одним. Причому деякі з них супідрядні іншим. Так, цілі є первинними по відношенню до всіх елементів, функції - визначають організовану форму їх реалізації, тобто структуру управління і т. д.

При проектуванні системи управління одним з найважливіших показників є кількість управляючої інформації (отже, і необхідна пропускна спроможність каналу). Розрахувати цей показник можна за допомогою інформаційно-ентропійного методу розрахунку.

Висновки до розділу

У цьому розділі було розглянуто основні системи управління телекомунікаційними мережами та їх стандарти, розкрито саме поняття системи управління, Також було описано принципи, на яких ці системи базуються та працюють.

2. РОЗРОБКА ОБ'ЄКТНО-ОРІЄНТОВАНОЇ МОДЕЛІ СИСТЕМИ УПРАВЛІННЯ ТЕЛЕКОМУНІКАЦІЙНОЮ МЕРЕЖЕЮ

2.1 Постановка задачі

Останнім часом розробниками все частіше вирішуються задачі проектування складних систем управління об'єктами інфокомунікацій.

Саме через це, об'єкти, яким поставлена задача створення нових концепцій при побудові систем управління (великих систем), представляють особливий інтерес. Піддамо аналізу виділяємі риси систем управління та їх моделей, що утворюються при проектуванні таких систем.

1. Здатність до еволюціонування в часі. Структура таких систем і їх функціонування не залишаються незмінними в часі. Нині телекомунікаційна мережа, яка виникла шляхом еволюції від розрізнених локальних мереж через об'єднання мереж в єдину інформаційну інфраструктуру, використовується не тільки за своїм прямим призначенням, але і є мережею передавання даних між обчислювальними машинами.

2. Наявність великої кількості суперечливих критеріїв управління. Як правило, ці критерії навіть не підлягають чіткому формулюванню. Це, насамперед, завадостійкість управляючої інформації, швидкість, затримка, вартість.

3. Присутність об'єктів управління (ОУ), поведінка яких значною мірою визначає поведінку всього об'єкта. На відміну від автоматів ОУ мають не жорстко детерміновану поведінку, а поведінку пов'язану з багатьма можливими режимами роботи. Це ускладнює прогнозування поведінки об'єкта управління, особливо тоді, коли є великий спектр зовнішніх впливів. У цьому разі доцільно використовувати вірогідність моделі прогнозування.

4. При побудові моделі системи управління в об'єктно-орієнтованому аналізі розглядали за початковий не унікальний об'єкт, а деякий типовий. Як тільки модель об'єкта (наприклад, у вигляді діаграми станів) була складена, її можна було використовувати для всіх об'єктів аналогічного типу. Модель управління

первинною мережею не може використовуватися при управлінні сучасною інфокомунікаційною мережею. Щоразу модель має складатися заново, оскільки в ній повинні відображатися всі особливості, властиві саме даному об'єкту. Крім того, внаслідок еволюціонування структури і функціонування об'єкта, потрібно мати модель відкритого типу, яка під час свого функціонування має збагачуватися і змінюватися.

2.2 Введення в об'єктно-орієнтовану методологію розробки систем

ОО технологія стрімко розвивається по всіх напрямках ЕОТ, як варіант вирішення задач, що пов'язані зі складністю утворюваних систем. Такий підхід не лише в програмуванні, а й в проектуванні UI, БД, мікроархітектур та навіть баз знань.

Актуальність саме такого всеохоплюючого підходу полягає в можливості застосування ОО у вирішенні різноманітного комплексу задач, що розв'язують проблеми складних систем. ОО базується на представленні того, що систему має сенс проектувати як множину взаємодіючих між собою об'єктів, та розглядати кожен із них, як представника певного класу, та слід зауважити, що вся ця множина буде утворювати ієрархію.

Підігрівання інтересу спеціалістів у сфері розробки до такого підходу зумовлено тим, що методика проектування та аналізу структури не гарантують зниження об'єму роботи та зменшення затрачених на цю роботу людино\годин. ОО підхід більше за інші підходить для опису справжнього, тобто, реального, процесу створення систем, і це не закінчується лише на програмних системах, який може відзначитись ітеративністю, та може бажати внести зміни до наразі готових і налагоджених рішень.

ООМ складається з:

- ОО аналізу;
- ОО проектування;
- ОО програмування.

Об'єктно-орієнтований аналіз

На цей підхід мали вплив минулі етапи розвитку ПЗ. На об'єктний підхід вплинули попередні етапи розвитку програмних засобів. Традиційні прийоми структурного аналізу засновані на потоках даних в системі.

ОО аналіз націлений на будівництво наближених до реальності моделей, з використанням ОО підходу; це методологія, при якій поняття об'єктів та класів, які становлять словник предметної області, формують вимоги.

За результатами цього аналізу вже можна формувати моделі, на яких базується ОО проектуванні; об'єктно-орієнтоване проектування являється основою остаточної реалізації системи з використанням методології ООП.

Головними перевагами ООМ порівняно зі структурними методами є:

- можливість подолати обмеження, пов'язані зі складністю розроблювальних систем;
- використання на стадії аналізу моделей близьких до реальності;
- застосування як при аналізі і проектуванні інформаційних систем, так і систем реального часу і апаратно-програмних комплексів;
- забезпечення можливості повторного використання розробленого програмного забезпечення, що дозволяє істотно скоротити терміни і знизити витрати на розробку кожної наступної системи;
- підтримка ітеративного, а не лавиноподібного, як у структурному підході, процесу проектування;
- природна робота з різномірною інформацією, що використовується в мультимедійних системах;
- створення більш відкритих систем;
- ціле використання ОО мов для опису.

Об'єктно-орієнтоване проектування

Методологія розробки об'єктної декомпозиції та спроектованої динамічної, статичної, логічної та фізичної моделі називають ОО проектуванням.

З цього можна виділити дві частини:

1. Декомпозиція набувається шляхом ОО проектування;

2. Для того, аби відслідкувати структуру (як логіку, так і фізику) моделі є необхідність використовувати різноманіття спеціальних прийомів уявлення.

Саме підтримка ОО декомпозиції має змогу відрізнити ОО проектування від структурного.

Об'єктно-орієнтоване програмування

ООП називають методику програмування, що ґрунтується на уявленні програми як множини об'єктів, які реалізовані за допомогою класів, а ці класи, відповідно, вибудовують певну ієрархію та спадкуються.

З цього можна виділити наступні правила:

1. ООП в своїй основі маніпулює не алгоритмами, а об'єктами;
2. будь-який об'єкт це лише готовий клас;
3. уся множина класів утворює ієрархію.

Основні положення такого підходу.

Можна виділити чотири пункти ОО методики:

- модульність
- ієрархія
- абстрагування
- обмеження доступу або інкапсуляція.

За відсутності хоча б одного з цих чотирьох пунктів модель перестає мати звання ОО. Також слід зазначити додаткові пункти:

- паралелізм
- типізація
- збереження.

Як можна зрозуміти з назви, ці пункти не обов'язкові, проте вони приносять значну користь моделі.

Абстрагування

Побороти складні системи призначений такий інструмент як абстракція. По суті, абстрагування це відволікання від точних характеристик у бік глобального значення об'єкту для виконання поставленого завдання. Якщо взяти, для прикладу, годинник, то можна сказати, що ми будемо обрати до уваги лише головну функцію, а саме «дізнаватися час», проте, ми не будемо обрати до уваги інші характеристики, такі як їх зовнішній вигляд, або фірма виробник цього годинника.

Таким чином можна дійти висновку, що абстрагування це відформаовані абстракції. Головна задача абстракції виділити для кожного об'єкту таку характеристику, яка відобразить його основу суть але у той же час надасть змогу відрізнити його від інших об'єктів.

Також абстракція не несе у собі задачі пояснити як об'єкт працює, з яких складових він складається, тощо. Вона лише допомагає зрозуміти об'єкт ззовні, легше за все абстрагувати, якщо виокремити задачу об'єкту.

Абстракція виокремлює значимі характеристики будь-якого об'єкту, такі, які допомагає зрозуміти його відміну від інших і таким чином дає спостерігачу зрозуміти його концептуальні кордони

Інкапсуляція

Інкапсуляцією можна назвати таку операцію, як відокремлення елементів, визначаючих поведінку, в розрізі одного об'єкту. Також потрібно додати, що вона призначена для відокремлення задач абстракції від реалізації цих задач.

Поняття інкапсуляції та абстракції це дві половини одного цілого. Перше працює з пристроєм, а друге слідкує за роботою. Інкапсуляція приховує деталі, які не мають задачі змінити поведінку таким чином, щоб її було видно зовні. Окрім цього приховується внутрішня структура і реалізація. Найчастіше інкапсуляція виконується за допомогою приховування інформації, тобто маскуванням всіх внутрішніх деталей, що не впливають на зовнішнє поведінку. Зазвичай ховаються і внутрішня структура об'єкта, і реалізація його методів.

Таким чином, інкапсуляція розмежовує абстракції.

Для того, щоб показати на наглядному прикладі приклад інкапсуляції, можна взяти будову рослини: аби дізнатись яким чином проходить фотосинтез на складних рівнях, зовсім необов'язково розуміти яким чином працює корінна система або яким чином побудована клітинна сітка даної рослини. Так само і при розробці БД програмна частина проектується таким чином, що вона ніяким боком не залежить від фізичних показників даної інформації, замість цього перевага віддається логічній схемі побудови. Для обох цих випадків задачі об'єкту надійно захищені від реалізації об'єктів нижнього рівня.

Задачею інкапсуляції є приховання реалізації.

Модульність

Модульністю можна назвати таку властивість системи, що дає можливість декомпонувати модулі, які слабо пов'язані між собою, проте внутрішньо пов'язані.

У класичному проектуванні поняття модульності описується як змogu розподілити підпрограми таким чином, що вони відсортовані по використанню змінної, або по використанню один одної. В ООП це трохи відрізняється – треба на фізичному рівні відділити об'єкти від класів.

Якщо брати невелику за обсягом задачу, можна допустити, що у одному модулі знаходяться описи об'єктів та класів. Проте, для більшої частини сучасних додатків розумніше буде мати декілька модулів, в кожному з яких будуть знаходитись об'єкти та класи які пов'язані логікою і у той же час, таким чином, щоб ті елементи, які повинні бути видимі для сусідніх модулів, були відкриті. На перший погляд така стратегія виглядає раціонально і правильно, проте на практиці може привести до неприємних наслідків. Для прикладу візьмемо задачу, яка потребує обладнання з множиною процесорів, а також яка потребує передачу повідомлень. Для величезних систем тисячі повідомлень не є чимось дивним. Наївно створювати модулі під кожен клас повідомлень, бо через це для користувача настають тяжкі часи, якщо він захоче провести пошук фрагментів.

Також при такому підході стає вкрай важким документування фрагментів. Як тільки в проекті рееструються зміни, проходить рекомпіляція і зміна сотні, якщо не тисячі модулів. Порозумівши такий феномен, можна дійти висновку, що іноді навіть повна відсутність модулів виглядає кращим рішенням, аніж програма розділена на модулі безсистемним чином.

Абстракції зберігаються окремо завдяки модульності.

Базуючись на практиці можна скласти список певних правил та методів, що в свою чергу дозволять найбільш ефективно створювати модулі з об'єктів та класів. Фінальною задачею розбиття ПЗ на модулі являється зниження необхідних ресурсів, як фінансових так і людських на його розробку, базуючись на незалежному тестуванні. Будова кожного модуля має бути проста для розуміння; реалізація одного модуля не має залежати від реалізації інших; є необхідність реалізувати модулі таким чином, щоб місця, де має вноситься більша частина змін були оптимізовані під цю задачу. Міркування ставлять межу цим керівним принципам. Проте якщо дивитись на практиці, то виходить, що насправді рекомпіляція тіла це не такий вже й складний процес, бо рекомпілюється тільки окремий модуль, а ПЗ перекомпонується. На практиці перекомпіляція тіла модуля не є трудомісткою операцією: заново компілюється тільки даний модуль, і програма перекомпонується. Однак рекомпіляція інтерфейсу навпаки – досить складна. Така рекомпіляція має таку структуру – рекомпіляція тіла і інтерфейса – рекомпіляція модулів, що пов'язані з даними – рекомпіляція модулів, що пов'язані зі вже рекомпільованими – далі, аж поки не рекомпілюються усі взаємозв'язки. Як результат, у великих системах це може зайняти великий проміжок часу (якщо тільки не присутня можливість фрагментарної компіляції) та, як наслідок, ресурсів, що недопустимо. Саме тому має сенс реалізовувати модулі таким чином, щоб абстракції були пов'язані логічно, а модулі мали мінімум взаємозв'язку.

Під час розподілення системи потрібно брати до уваги та використовувати два простих правила. Перше – розподіл повинен мати на увазі, модулі це неподільні,

елементарні блоки, які можуть використовуватись не один раз. Друге – переважна більшість компіляторів для кожного модуля роблять окремий сегмент коду, через це може бути обмеження на об'єм модулю. Потрібно не забувати про те, що кількісь викликів підпрограм, які всередині модуля може сильно вплинути на віртуальну пам'ять, точно так само як і розташування описів. При невірному розподілі процедур, створюється більше взаємозв'язків між сегментами, що в свою чергу відображається на втраті ефективності.

При обранні що саме розбивати беруться до уваги також зовнішні обставини. При розробці у групі людей зазвичай робота розподіляється у вигляді модульного розподілу і таким чином мінімізується взаємозв'язок між членами групи. Як показує практика, реалізацією займаються фахівці з малим досвідом роботи, а інтерфейсом навпаки – більш досвідчені. Якщо брати на розгляд більш серйозний рівень, то такий підхід також справедливий, проте будується вже для субпідрядників. Нащастя, є можливість розподілити абстракцію таким чином, що інтерфейс модуля можна встановити досить швидко, лишається тільки дійти згоди компаніям, які займаються розробкою. Мінімалізму інтерфейсу також можна дати розумне пояснення – кожна його зміна веде за собою тонни витраченого паперу, багато галасу, витрачені нерви. Документообіг також реалізується по принципу модульності. В такій реалізації опис та адміністрування це один модуль. Проте тут є підводні камені. Десятки або сотні описів представляють собою, відповідно, десятки або сотні модулів, що негативно впливає на декомпозицію. Можна запровадити рівні безпеки (секретності) та деякі, важливі частини коду приховувати, даючи доступ тільки санкціонованим працівникам, а незначні напрацювання, що не несуть глобальної цінності лишати загальнодоступними. При такому впровадженні секретна частина також представляється у вигляді одного або декількох модулів.

Зібрати до купи усі ці вимоги, що суперечать одне одному досить непросте завдання, проте, є необхідність зрозуміти: розмежування об'єктів та класів, під час розробки проекту це дії, які не залежать один від одного. Сам цей процес є

логічним проектуванням, а розділення системи на модулі це вже фізичне проектування. Зрозуміло, що бувають такі ситуації, коли фізичну частину не можна завершити без логічної або навпаки. Обидва ці процеси виконуються у прямій залежності один від одного.

Ієрархія

Говорячи про абстракцію, можна зрозуміти, що це дуже корисна штука, проте в розумних межах. Однак іноді число цих абстракцій перевищує допустиму кількість. Інкапсуляція, працюючи разом з абстракцією, допомагає вирішити цю проблему за допомогою приховання змісту абстракції з поля зору. Модульність, разом з ними, допомагає вирішувати проблеми несумісності та поєднує абстракції, що зв'язані логікою у групи. Але як показує практика, весь цей спектр дій іноді все одно не задовольняє потреби у повній мірі.

Спрощення задач відбувається за рахунок опису ієрархічних структур абстракцією. Можна виділити ієрархію. Визначимо ієрархію наступним чином:

Сортировка абстракції та розміщення їх по рівнях є ієрархією.

Стандартні види структур у складних системах мають вигляд для класів - ієрархія "is-a", для об'єктів - ієрархія "part of"

"Is-a" базується на спадкуванні. Спадкування можна пояснити як запозичення одним класом характеристик, функцій, структури у іншого(інших) класів. Тобто, можна сказати, що спадкування створює ієрархію таким чином, що в ній підкласи запозичують будову у суперкласів. На практиці видно, що часто підклас, що успадкував якусь частину, перебудовує у кращий бік цю частину у вищого за нього класу.

Якщо спробувати пояснити роботу "is-a", то можна, наприклад, сказати що ведмідь це савець, а будинок це нерухомість. Наприклад, ведмідь є савець, будинок є нерухомість. Виходить, що таке наслідування створює ієрархію "узагальнення-спеціалізація", таку, що в ній підклас це окремий випадок суперкласу. У той же час ці суперкласи показують загальні, а підкласи –

спеціалізовані абстракції. У таких спеціалізованих абстракціях частини суперкласу є змога модифікувати, приховати або доповнити. Такий принцип допомагає зробити проєкт менш громіздким.

Ієрархія простого успадкування

Інший різновид ієрархічної організації - " part of "- ієрархічна структура - базується на щодо агрегації. Агрегація не є поняттям, унікальним для об'єктно-орієнтованих систем. Наприклад, будь-який мова програмування, що дозволяє структури типу "запис", підтримує агрегацію. Проте в ООП вона відкриває для себе нову можливість: за допомогою агрегації структури, що пов'язані логікою фізично групуються, а потім спадкуються в різні абстракції.

Типізація

Спосіб захисту та керування об'єктами від їх підміни називають типізацією.

Вона примушує описувати абстракції таким чином, що при програмуванні є змога дотримуватись потрібних рішень.

Концепція базується на абстракції даних. Тип - точне визначення властивостей будівлі або поведінки, яке властиве певної сукупності об'єкта. Часто терміни "тип" і "клас" вважають еквівалентними. Більш точно сказати, що клас реалізує тип.

Об'єктно-орієнтовані мови програмування можуть бути: суворо типизованими, не суворо типизованими і зовсім не типизованими, що дозволяє говорити про типізації, як про другорядний елементі. Сильно типізовані мови - це такі мови, в яких всі вирази проводять перевірку на відповідність типів.

Розрізняють статичну типізацію (раннє зв'язування) і динамічну типізацію (пізніше зв'язування). Поділ має відношення до часу, коли імена пов'язують з типами: статична - під час компіляції; динамічна - під час виконання програми.

Поліморфізм

Поліморфізм виникає на стику принципів успадкування і динамічних зв'язків. Ця властивість є найбільшим в об'єктно-орієнтоване програмування. Поліморфізм відрізняє об'єктно-орієнтоване проектування від більш традиційних методів з використанням абстрактних типів даних.

Поліморфізм - можливість за допомогою одного імені позначати операції з різних класів (але що відносяться до загального суперкласу).

Можливість заміни один одним об'єктів, які мають однаковий інтерфейс називають поліморфізмом.

Щоб дізнатися, чи можна використовувати поліморфізм у тій чи іншій мові програмування, треба перевірити чи можна змінити реалізацію класу при спадкуванні. Суть поліморфізму можна виразити таким чином – множина реалізацій за допомогою єдиного інтерфейсу.

За допомогою використання поліморфізму є змога створювати абстрактні програми, а також збільшувати коефіцієнт повторного використання. Усі характеристики кожного об'єкту складаються у інтерфейс, клас. Те, що спільне можна прослідкувати як внутрішньо так і зовнішньо. Зовні це можна простежити за методами, що мають однакову сигнатуру та називаються однаково. Зсередини ж можна зрозуміти що у всіх цих методів однакоий функціонал.

Реалзувати поліморфізм можна за допомогою наслідування. Клас, що наслідується передає свої сигнатури класу, що наслідує, проте, реалізація в них може бути не схожа. Таке явлення прийнято називати перевизначенням. Також буває, що працює функціонал батька, проте під час виконання роботи буде підставлятися нащадок. Таке поняття називають тіснозв'язуванням.

Паралелізм

Для деякої категорії задач АС виконують обробку більшості івентів, що працюють в один і той же час. Одночасно з тим, як ООП базується на

інкапсуляції, спадкуванні та абстракції, паралелізм тісно зв'язаний з абстрагування та синхронізацією процесів.

Основою, що об'єднує ці концепції є об'єкт. Кожен із них (який можна представити абстракцією реальності) може представляти собою виділений потік управління.(абстракцію процесу). Такий вид об'єкту називають активним.

Властивість відрізняти пасивні та активні об'єкти називають паралелізмом.

Для систем, що будують на основі ООП, реальність можна представити, як множину об'єктів, що взаємодіють, деякі з яких – активні.

В ООП існує декілька підходів до паралелізму:

1. Паралелізм - це внутрішня властивість деяких мов програмування. Для прикладу, у мові програмування Ada паралельні процеси виконуються як завдання. Якщо брати SmallTalk, то в ньому є клас, що називається Process, який усі активні об'єкти успадковує. Існують і інші мови програмування з вбудованими механізмами, що дозволяють виконувати паралельно та синхронізувати процеси, такі як Actors, Orient 84/K, ABCL/1, що передбачають подібні механізми паралелізму і синхронізації. Для всіх цих мов існує можливість створення активних об'єктів, код яких постійно виконується одночасно та паралельно з іншими активними об'єктами.

2. Є змога використання бібліотеки класів, що дозволяють реалізувати полегшено паралелізму. Наприклад, бібліотека AT&T для C++ містить класи Shed, Таймер, Task і т.д. Реалізація цієї бібліотеки залежить виключно від платформи, проте слід зазначити що її інтерфейс дуже добре переноситься. Використовуючи такий підхід, паралельне виконання не вбудовується саме в мову та не впливають на інші системи, що працюють без паралельності, але сприймаються все ж таки як вбудовані.

3. За допомогою переривань можна створити ілюзію багатозадачності, проте для цього потрібно виконати деякі маніпуляції з апаратною частиною.

Одразу після введення паралелізму постає проблема, як синхронізувати активні об'єкти, а також з іншими послідовними об'єктами. Наприклад, якщо два

об'єкта почнуть одночасно посилати команди третьому, то потрібно бути впевненим в тому, що об'єкт, який отримує дві команди одночасно не зруйнується через таку операцію зміни його стану. Тут вже поєднуються інкапсуляція, паралелізм та абстракція. У таких системах потрібно не лише визначити реакцію об'єкту, а й вжити ряд мір, які не допустять руйнування цього об'єкта на частини через запит від кількох незалежних процесів.

Зберігання /стійкість

Зберігання/стабільність – можливість об'єкту існувати не озираючись на те, хто був його творцем.

Можна відзначити такі стійкі види об'єктів:

- результати обчислення
- змінні процедур
- глобальні змінні
- дані основної програми
- загальні дані для всіх версій програми
- дані, які залишаються навіть після того, як програма, що їх створила вже не працює

Як можна помітити, перші три пункти виконуються за допомогою програмування, а останні три за допомогою БД. Таким чином ми наближаємось до ОО БД.

2.3 Аналіз вимог і визначення специфікацій програмного забезпечення при об'єктному підході

Розроблюємі за допомогою об'єктного підходу моделі ПЗ, тримаються на явищах та предметах реального світу. У їх основі закладена необхідна реакція ПЗ, тобто виконання покладених на нього функцій, але вона пов'язана з положенням елементів конкретної предметної області.

Виходить, етап аналізу ставить такі задачі:

- створити концептуальну модель предметної області спираючись на задачі;
- більш детально зрозуміти реакцію створюваного ПЗ;

UML - стандартна мова опису розробки програмних продуктів з використанням об'єктного підходу.

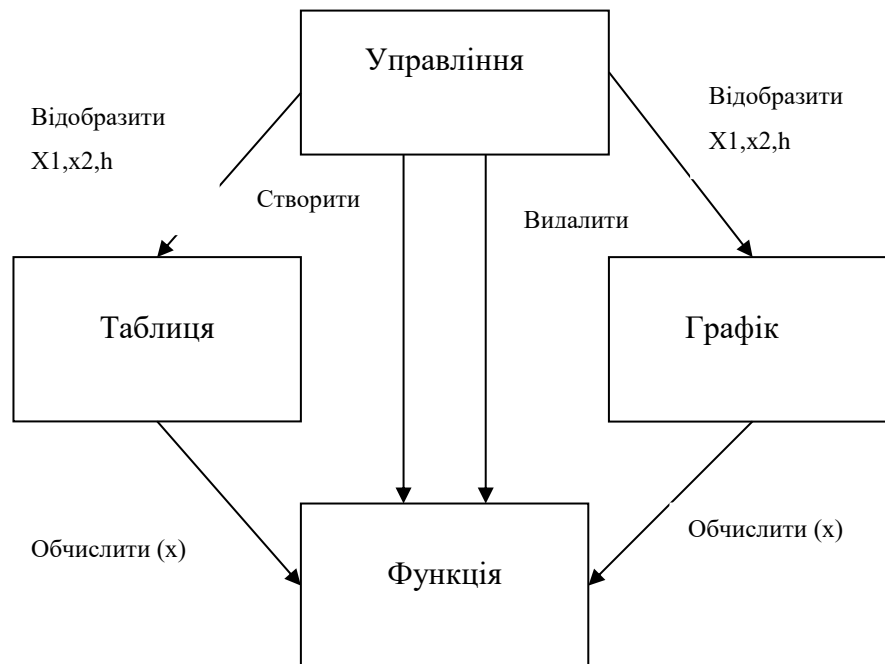


Рисунок 2.1 - Об'єктивна декомпозиція програми побудови таблиць і графіків

Фундаментом об'єктної розробки ПЗ є об'єктна декомпозиція. Це подання розробляемого програмного забезпечення як множини об'єктів, що при взаємодії та передачі повідомлень винокує поставлені функції. (рис.2.1).

При такому підході декомпонувати можна лише просте ПЗ. Саме через те були запропоновані ще й інші методи розробки та аналізу забезпечення в рамках цього підходу, використовуючи різні моделі. Сперечатися про переваги і недоліки цих методів і моделей можна було нескінченно. Ця ситуація отримала назву «війни методів».

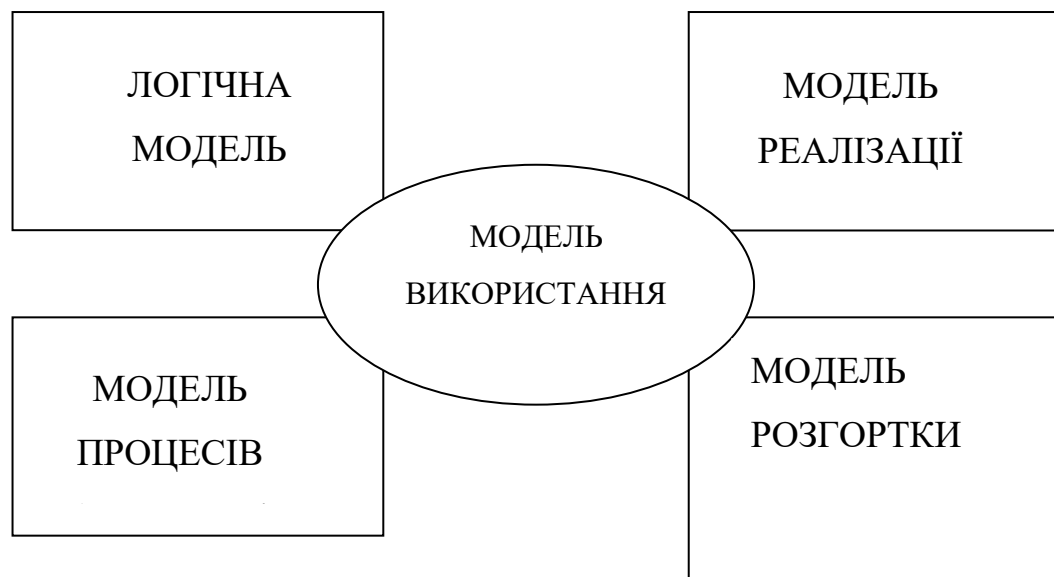


Рисунок 2.2 - Повна специфікація ПЗ, що розробляється, при об'єктному підході

Кінець цієї війни прийшовся на 1995 рік, через появу першої версії уніфікованої мови моделювання, тобто UML. У сучасному світі вона рахується як стандартний засіб описів об'єктно-орієнтованих проєктів. Її, можна сказати, створили декілька спеціалістів світового рівня у області моделювання, такі як Д. Рамбо, І. Якобсон та Г. Буч. Кожен із них додав у цю мову краще своє напрацювання. Специфікація ПЗ при використанні цієї мови поєднує у собі такі моделі: використання, логічний, реалізації, процесів, розгортання (рис.2.2).

Моделлю використання можна назвати опис функцій ПЗ

Абстракцію ПЗ описує модель логіки. Вона пояснює що потрібно для підтримку достатньої функціональності.

Організація модулів визначається моделлю реалізації.

Оцінка продуктивності і надійності виконується за допомогою моделі процесів.

Зрозуміти, як розміщено компоненти на обладнанні призначена модель розгортання.

Виходить, що всі ці моделі, які описані, розкривають різні частини одної системи у розрізі, а разом дозволяють скласти загальну картину того забезпечення, що розробляється.

Пропонується дев'ять діаграм, які доповнюють одна одну, проте стосуються різних моделей: класів, використання, кооперації, діяльність, пакети, послідовності, стан, компоненти, розміщення.

Всі вони виконані в єдиному стилі. Це зроблено задля того, щоб була змога максимально швидко їх зрозуміти.

Окрім цього існує словник термінів. Конкретний перелік цих термінів описується кожним розробником UML. Побудувати UML можливо за допомогою Microsoft Visual Modeler. Статистика показує, що майже всі провідні компанії використовують UML при розробці, що говорить про те, що цей стандарт став еталоном.

2.4 Розширення функціональних можливостей

Для додавання до варіанту використання нових дій зручно застосовувати ставлення розширення. З його допомогою базовий варіант використання може бути розширений новим варіантом використання.

У нашому прикладі поведінку системи не визначено для випадку, коли що здається елемент застряг. Введемо варіант використання Елемент Застряг, який буде розширювати базовий варіант використання Повернення Елемента (рис. 2.3).

Опис варіанти використання Елемент застряг може мати наступний вигляд: Якщо елемент застряг, для виклику Оператора виробляється сигнал тривоги. Після видалення застряглого елемента Оператор скидає сигнал тривоги. У результаті Споживач може продовжити задачу елементів. Величина РАЗОМ зберігає правильне значення. Ціна застряглого елемента не зараховується.

Таким чином, опис базового елемента залишається колишнім, простим. Ще один приклад наведено на рис.2.4 Тут ми бачимо тільки один базовий варіант використання Сеанс роботи. Всі інші варіанти використання можуть додаватися

як розширення. Базовий варіант використання при цьому залишається майже без змін.

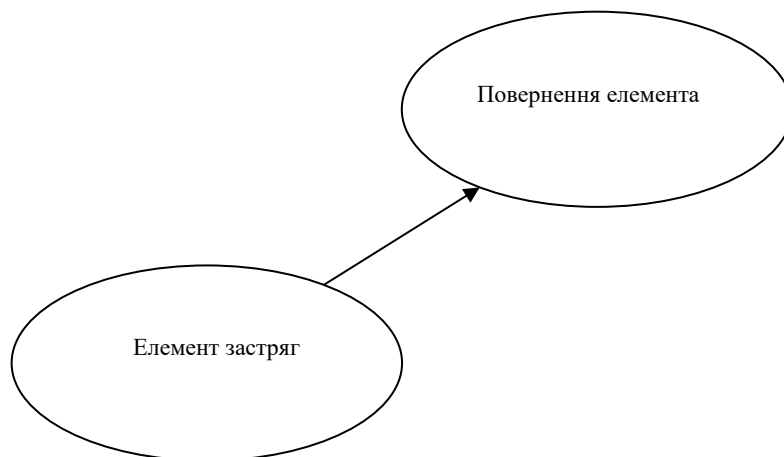


Рисунок 2.3 - Розширення варіанта використання Повернення елемента

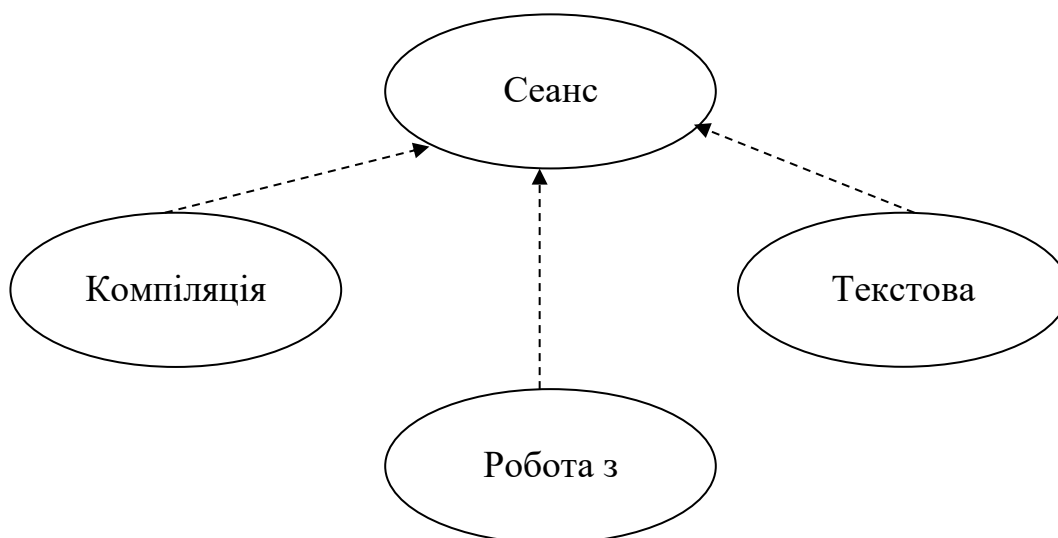


Рисунок 2.4 - Використання відносин розширення

Ставлення розширення визначає переривання базового варіанта використання, яке відбувається для вставки іншого варіанту використання. Базовий варіант використання не знає, буде виконуватися переривання чи ні. Обчислення умов переривання знаходиться поза компетенцією базового варіанта використання. У розширюючому варіанті використання вказується посилання на те місце базового варіанта використання, куди він буде доданий (при переривання). Після

виконання розширюючого варіанти використання продовжується виконання базового варіанта використання. Зазвичай розширення використовують: для моделювання варіативних частин варіантів використання; для моделювання складних і рідко виконуваних альтернативних послідовностей; для моделювання підлеглих послідовностей, які виконуються тільки в певних випадках; для моделювання систем з вибором на основі меню.

Головне, що слід пам'ятати: рішення про вибір, підключенні варіанти на основі розширення приймається поза базового варіанта використання. Якщо ж ви вводите в базовий варіант умови умовну конструкцію, конструкцію вибору, то доведеться застосовувати ставлення включення. Це випадок, коли «штурвал управління» знаходиться в руках базового варіанта використання. Приклад найпростішої діаграми випадків використання, в якій використані відносини розширення, наведено на рис. 2.5.

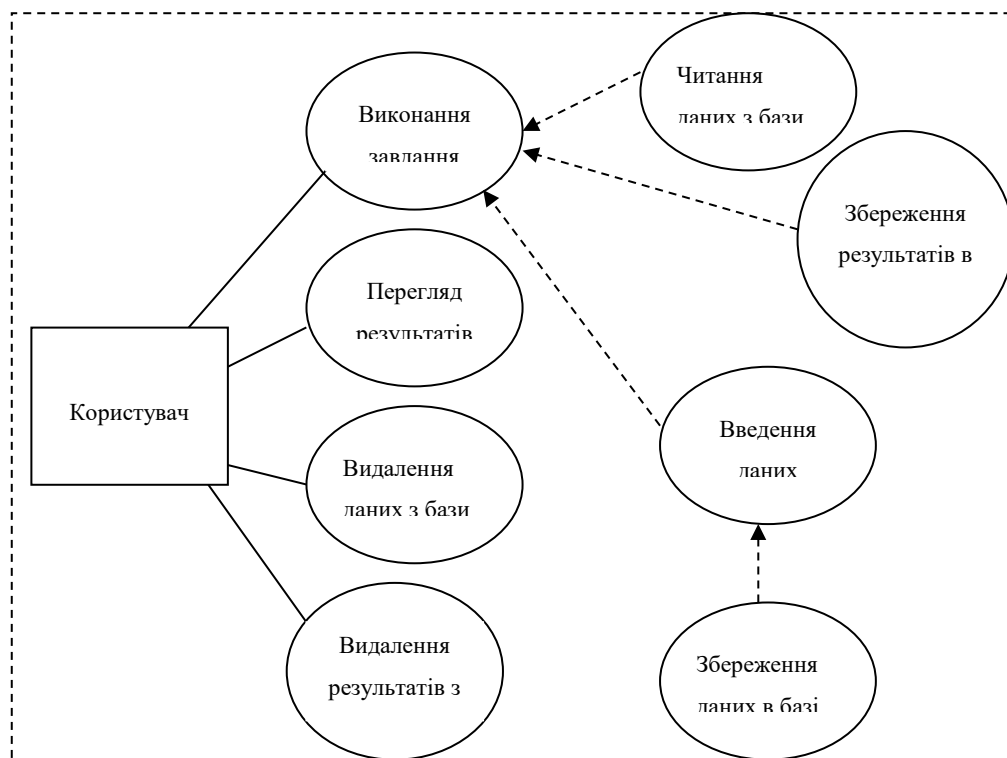


Рисунок 2.5 - Діаграма варіантів використання для системи рішення комбінаторно-оптимізаційних задач.

Як показано на рис.2.5 варіант використання «Виконання завдання» розширюється варіантами використання «Читання даних з бази», «Введення

даних», «Збереження результатів в базі», які визначають спосіб отримання даних (ввід з клавіатури або читання з бази) та збереження результатів в базі. «Варіант використання» «Введення даних» розширюється варіантом використання «Збереження даних у базі», який визначає зберігати чи ні дані у базі.

Уточнення моделі вимог

Уточнення моделі зводиться до виявлення однакових частин у варіантах використання і вилучення цих частин. Будь-які зміни в такій частини, виділеної в окремий варіант використання, будуть автоматично впливати на всі варіанти використання, які використовують її спільно.

Отримані варіанти використання називають абстрактними. Вони не можуть бути конкретизовані самі по собі, застосовуються для опису однакових частин в інших конкретних варіантах використання. Таким чином, описання абстрактних варіантів використання використовуються в описах конкретних варіантів використання. Кажуть, що конкретний варіант використання знаходиться відносно «включає» з абстрактним варіантом використання.

Повернемося до нашого прикладу. У цьому прикладі два конкретні варіанти використання Повернення елемента і Створення денного звіту мають загальну частину - дії, що забезпечують друк квитанції. Тому, як показано на рис. 6, можна виділити абстрактний варіант використання Друк. Цей варіант використання спеціалізуватиметься на виконанні роздруківок. У свою чергу, абстрактні варіанти використання можуть використовуватися іншими абстрактними варіантами використання. Так утворюється ієрархія. При ієрархії абстрактних варіантів використання керуються правилом: виділення варіантів використання припиняється при досягненні рівня окремих операцій над об'єктами. Виділення абстрактних варіантів використання можна спростити з допомогою абстрактних акторів.

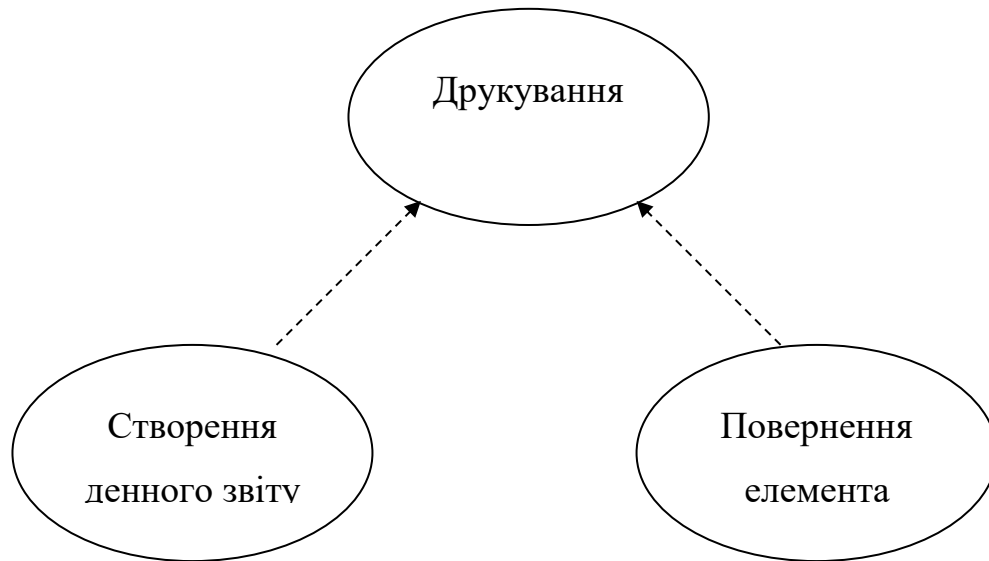


Рисунок 2.6 - Застосування відносини включення

Абстрактний актор - це загальний фрагмент ролі в декількох конкретних акторів. Абстрактний актор висловлює подібності в варіантах використання. Конкретні актори знаходяться щодо спадкування з абстрактним актором.

2.5 Порядок виконання процесів

ДПДД виражає вимоги на порядок виконання за допомогою визначень доступності введів. Вимоги пиводяться як правила виконання для ДПДД:

Процес може виконуватися, коли всі вводи доступні.

1. Висновки процесу доступні після того, як процес завершує виконання.

2. Дані подій (атрибути на потоках дних подій) завжди доступні.

3. Дані з архівів даних і зовнішні сутності завжди доступні.

4. Правила виконання забезпечують наступний алгоритм для вионання ДПДД з використанням маркерів. Спочатку розміщуєм маркери на всі потоки даних, які утримують доступні дані: усі потоки даних подій та усі потоки, що виникають в архівних даних та зовнішніх сутностях.

Повторюємо до тих пір, дои дія не завершена.

Якщо існує, але ще не виконаний процес, в якому всі виводи доступні, виконуємо процес, розміщуючи маркери на всіх виводах. Якщо не існує такого процесу, дія завершена.

При оцінці порядку виконання процесів у межах дії потік управління розглядається подібно потоку даних, який не несе ніяких даних. Це може бути виражено більш явно повторним приведенням двох правил виконання:

Процес може виконуватися, коли усі вводи (включаючи вводи управління) доступні.

Виводи процесу (включаючи виводи управління) доступні після того, як процес завершує виконання.

Багатократне використання процесів

Завдяки способу, за допомогою якого процеси в ООА розгалужуються (спочатку для моделей станів і потім для дій в межах моделі станів), бачимо, що один і той самий процес використовується в декількох ДПДД або в межах однієї моделі станів, або в декількох модлях станів. Оли процес багатократно використовується в цьому сенсі, він повинен бути відмічений одним і тим ідентифікатором процесу та ім'ям, де б він не з'явився на ДПДД.

Критерії багатократного використання

Щоб виявити, які два процеси є однаковими (тобто який процес багатократно використовується в том самому вигляді), перевіряють, які саме два процеси:

- виконують ту ж саму функцію;
- читають (або записують) ті ж самі атрибути з одних і тих самих архівів даних;
- приймають як ввід ті ж самі атрибути з джерел, які відрізняються від архівів даних;
- створюють як виводи одні і ті ж самі атрибути, які використовуються іншими процесами;
- створюють одні і ті ж події як виводи;
- створюють одні і тіж виводи умовного управління.

Вводи управління і виводи безумовного управління не мають відношення до визначення багатократного використання процесів, так як вони діють тільки тоді, коли процес вконується, і не є внутрішніми діяльностями процесу.

Формування та визначення процесів

В ООА процес включає в себе як обчислення або перетворення даних, так і будь-яку роботу, необхідну для зчитування або запису даних в архив даних. Процес не повинен робити якусь особливу роботу для доступу до даних, які представлені або подією, або іншим процесом.

Дію потрібно розрізнати для отримання процесів чотирьох визначених типів:

- аксесори;
- генератори подій;
- перетворювачі;
- перевірки.

Ці типи характеризуються в термінах мети процесу та використання архивів даних об'єкту.

Аксесор – процес, мета якого полягає в тому, щоб отримати доступ до даних одного архиву даних об'єкту. Аксесор може бути одним з наступних типів.

Аксесор створення – створює новий екземпляр об'єкту.

Аксесор читання – зчитує атрибути одного об'єкту.

Аксесор запису – модифікує атрибути одного об'єкту.

Аксесор знищення – видаляє екземпляр об'єкту.

Аксесор визначається для об'єкту, відповідного архиву даних, до якого він отримує доступ. Зазвичай аксесори – найбільш багатократно використовувані процеси як у межах дій однієї моделі стану, так і в діях різних моделей станів.

Генератори подій – це процес, який створює лише одну подію як вивід.

Генератор подій визначається для об'єкту, до якого направлена вихідна подія.

Перетворення – це процес, метою якого є обчислення або перетворення даних: процес існує щоб перетворювати свої вхідні дані в нову форму, яка потім стає виводом.

Перетворення визначається для об'єкту , відповідної моделі станів, в яку воно вкладене, і може зчитувати або записувати об'єктний архів даних об'єкту, для якого воно призначене.

Перевірка – це процес, який перевіряє умови та робить один з декількох умовних виводів управління. Перевірка визначається для об'єкту, який відповідає моделі станів, в яку вона вкладена, і можливе зчитувати або записувати об'єктний архів даних об'єкту, для якого воно призначене.

Перевірка – процес, який перевіряє умови і дає один з декількох умовних виводів управління. Перевірка визначається для об'єкту, який відповідає моделі станів, в яких вона вставлена, і може зчитувати (або записувати) атрибути об'єкту, для якого вона визначена. Перетворення і перевірки зазвичай найбільш близько зв'язані з внутрішніми методами в ООД.

2.6 Інформаційна модель

Як вже зазначалося раніше, ООА передбачає розділення системи на окремі, незалежні сутності. При попередньому перегляді системи управління ділянки об'єкт управління (ОУ) – вузловий пункт управління (ВПУ) маємо три об'єкта:

ОУ; 2 ВПУ; 3. З'єднувальна лінія, яка з'єднує ОУ та ВПУ, при цьому цей об'єкт враховує усю апаратуру, яка забезпечує зв'язок.

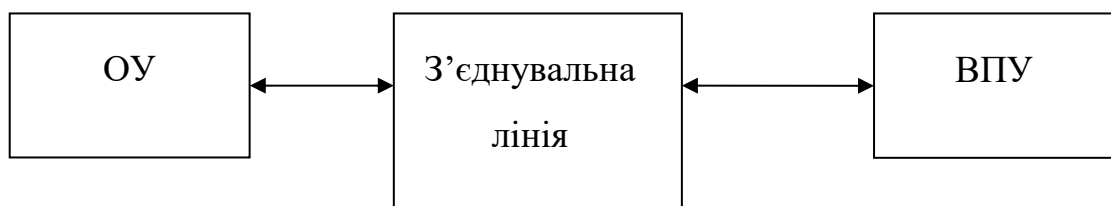


Рисунок 2.7 - Інформаційна структура

Для початку, розглянемо кожний об'єкт окремо та виділимо особливості, які характеризують тільки цей клас об'єктів. Так для ВПУ маємо:

(ВПУ) - вузловий пункт управління
(TDirectPoint) DirectPoint_ID – номер даного ВПУ Name - назва

Таким чином ВПУ має власний номер, і назву, яке може використовуватися як короткий опис даного ВПУ.

ОУ можна описати таким чином:

(ОУ) - об'єкт управління CtrlObj
CtrlObj_ID – ідентифікатор ОУ DirectPoint_ID – ідентифікатор ВПУ, якому належить даний ОУ Name – назва State – поточний стан Errors – ймовірність відмови

ОУ, крім свого ідентифікатора, ідентифікатора ВПУ та ім'я, характеризується ймовірністю відмови і поточним станом, які залежать від того, що імітує даний CtrlObj (вузол апаратури або транспортна мережа). Наприклад, якщо це вузол – то поле State може приймати значення «Норма», «Аварія», «Попередження» та «Ушкодження», або, якщо це цифровий комутатор каналів/трактів до цих значень додається ще і «Перемкнений» і т.і. Однак стан об'єктів буде розглядатися ширше при побудові моделей станів.

З'єднувальна лінія зв'язує ОУ та ВПУ, тобто це те за допомогою чого ОУ спілкується з ВПУ – передає повідомлення і отримує дані. Під лінією будемо розуміти самотестуючий та незалежний комплекс пристроїв, який забезпечує передачу інформації по фізичній лінії або каналу (наприклад локальна мережа або телефонний канал). В цьому комплексі звертаємо увагу тільки на його надійність.

(TCommChanel) - з'єднувальна лінія
ComCH_ID

ID_DirectPoint
ID_InfoPoint_ID
State
Erorrs

З'єднувальну лінію характеризують: ідентифікатори ОУ та ВПУ, стани та ймовірність відмови.

Опираючись на характеристики об'єктів, а також з урахуванням топології ІМ цієї ділянки можливо побудувати інформаційну модель системи управління інфокомунікаційною мережею (рис.2.8)

Мета етапу інформаційного моделювання полягає у тому, щоб ідентифікувати концептуальні сутності, або об'єкти, які складають підсистему для аналізу.

Об'єкти зображуються на інформаційній моделі разом з характеристиками або атрибутами. Зв'язки, які властиві об'єктам, представляються на графічній моделі як з'єднання між об'єктами. Завершений опис ви визначення кожного об'єкту, атрибуту та зв'язку повинно бути підготовлено як документація для графічної моделі.

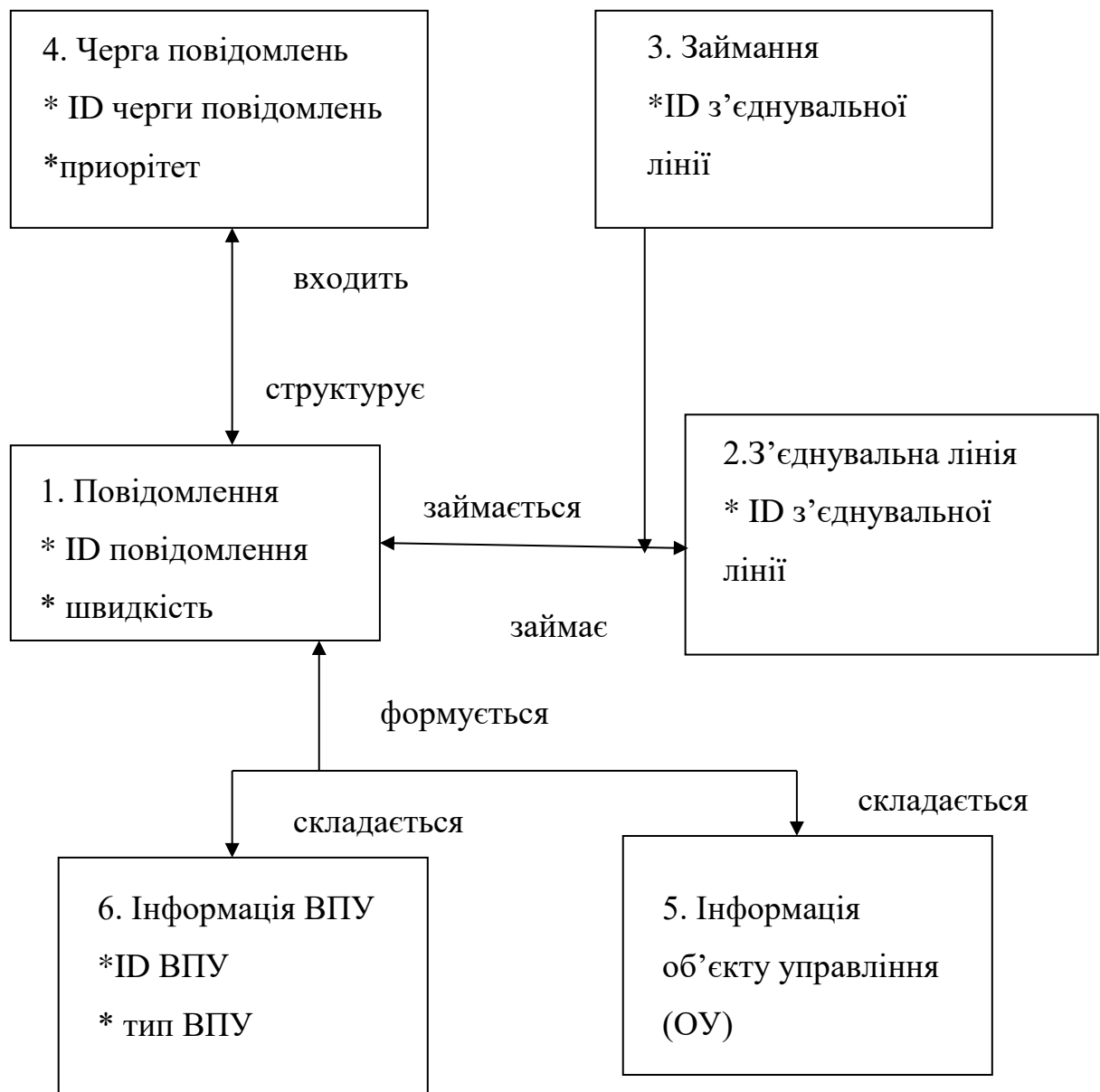


Рисунок 2.8 - Інформаційна модель системи управління інфокомунікаційною мережею

Висновки до розділу

У цьому розділі було здійснено введення в об'єктно-орієнтовану методологію розробки систем. Проведено об'єктно-орієнтований аналіз та проектування. Розкрито поняття інкапсуляції, модульності, ієрархії та інших. Виконано аналіз

вимог і визначення специфікацій програмного забезпечення при об'єктному підході. Зазначений порядок виконання процесів

3. ПОБУДОВА ОБ'ЄКТНО-ОРІЄНТОВАНОЇ МОДЕЛІ СИСТЕМИ УПРАВЛІННЯ ТЕЛЕКОМУНІКАЦІЙНОЮ МЕРЕЖЕЮ

3.1 Побудова концептуальної моделі предметної області

Щоб побудувати концептуальну модель необхідно звернутися за допомогою до діаграми класів. Такі моделі використовують предметну область, їх атрибути та відносини. Під поняттями можна розуміти не тільки фізичні предмети, а й абстракції, що використовуються фахівцями.

Основою будь-якої моделі є клас. При цьому, сам клас описується як множину ознак групи об'єктів. Беручи це до уваги, зрозуміло, що на діаграмі до кожного класу відповідає множина об'єктів. Для прикладу, клас Студент включає всіх студентів. Проте кожен екземпляр володіє своєю сукупністю характеристик, навіть таких, які у самому класі не описані. Як прикладу, окремий студент може бути ще й спортсменом. Таким чином приходимо до висновку, що головним ідентифікатором є ім'я.

Практика показує, що зазвичай поняття на діаграмі подаються таким чином:

- створюється множина понять, які надають характеристику предметній області;
- поняття, що малозначимі для використання виключаються.

Для того, щоб визначити цю множину понять, рекомендовано орієнтуватись на перелік понять, які наведені у таблиці 3.1

Побудуємо концептуальну модель для системи рішення комбінаторно-оптимізаційних задач (діаграма варіантів використання на рис.3.1). Безліч понять кандидатів для даної розробки включає наступні словосполучення:

Завдання, тип завдання, список типів завдання, спосіб завдання даних, введення даних, вибір даних з бази, алгоритм розв'язання задачі, повнота описи, результати, дані, база даних.

Спробуємо виділити основні поняття і пов'язати їх між собою.

Мета основного варіанти використання системи - виконання завдання. Повний опис завдання включає: тип завдання, дані і вказівка на алгоритм. З ним же будуть пов'язані і отримані результати. Дані можуть зберігатися в базі і вводиться. Опис завдання і все, що з ним пов'язано, може зберігатися в базі.

Визначимо можливі узагальнення:

- спосіб завдання даних: введення даних, вибір даних з бази.
- алгоритм: алгоритм розв'язання задачі: конкретний алгоритм розв'язання задачі. Переходимо до побудови концептуальної моделі.

Категорія понять-кандидатів	Приклади
Фізичні або матеріальні об'єкти	Система управління - як ціле
Специфікації, елементи дизайну або опису об'єкту - зберігаються в системі, навіть при відсутності об'єктів	Конфігурація, параметри, функції
Місце	Інфокомунікаційна мережа
Роль людини	Отримувач послуг, адміністратор мережі, абонент
Контейнери інших об'єктів	Система управління - як сукупність частин, параметри - як сукупність описів
Вміст контейнерів	Частина, елемент
Інші комп'ютери або зовнішні системи по відношенню до системи, що розробляється	Система транспортування інформації
Абстрактні поняття	Управляюча інформація
Організації	Фірма, підприємство
Події	Передача управляючої інформації, передача даних
Процеси та їх частини (представляються у вигляді класу, якщо не аналізуються елементи процесу)	Очікування, прийом повідомлення, передача повідомлення
Правила і політика	Правила роботи системи управління
Записи фінансової, трудової, юридичної та іншої діяльності, керівництва, книги	Книга обліку, посадова інструкція

Таблиця 3.1 - Перелік можливих категорій понять-кандидатів

Основний клас-поняття, виходячи з опису, Завдання. Пов'язуємо класи поняття Дані, Алгоритм і Результати.

У системі протиракетної планується реалізувати алгоритми рішення задач трьох типів: пошук циклу мінімальної довжини, що проходить через всі вершини; пошук найкоротшого шляху пошук мінімального покривного дерева. Отже, клас-поняття Алгоритм є суперкласом для класів Алгоритм пошуку циклу мінімальної

довжини, Алгоритм пошуку найкоротшого шляху і Алгоритм пошуку мінімального покривного дерева (рис.3.1). Від яких, у свою чергу, будуть успадковуватися Алгоритми, які реалізують конкретні методи. Алгоритм також пов'язаний з Даними і Результатами.

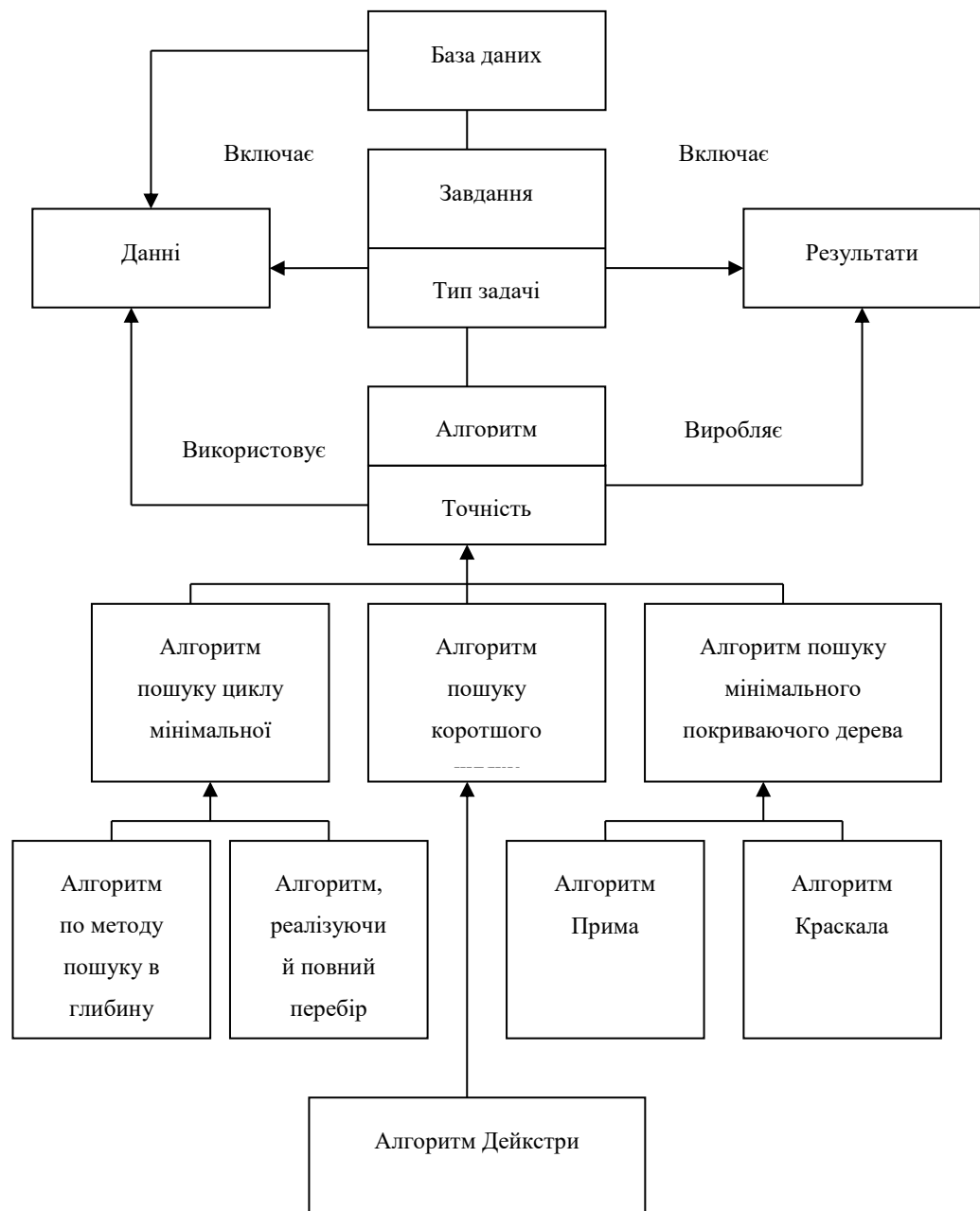


Рисунок 3.1 - Контекстна діаграма класів для системи рішення комбінаторно-оптимізаційних задач.

Концептуальна модель характеризує статичні властивості ПО, що розробляється. Для опису особливостей його поведінки, тобто можливих дій системи доцільно використовувати: діаграми послідовностей системи, діаграми діяльності і діаграми станів об'єктів.

3.2 Динамічні, статичні моделі та моделі реалізації об'єктно-орієнтованих програмних систем

Основним завданням логічного проектування при об'єктному підході є розробка класів для реалізації об'єктів, отриманих при об'єктній декомпозиції, що передбачає повний опис полів і методів кожного класу.

Фізичне проектування при об'єктному підході включає об'єднання класів та інших програмних ресурсів у програмні компоненти, а також розміщення цих компонентів на конкретних обчислювальних пристроях.

Статичні моделі об'єктно-орієнтованих програмних систем забезпечують подання структури систем у термінах базових будівельних блоків і відносин між ними. «Статичність» цих моделей полягає в тому, що тут не показується динаміка змін системи в часі. Разом з тим слід розуміти, що ці моделі несуть у собі не тільки структурний опис, але і опис операцій, які реалізують задану поведінку системи. Основним засобом для подання статичних моделей є діаграми класів.

Динамічні моделі забезпечують подання поведінки систем. «Динамізм» цих моделей полягає в тому, що в них відображається зміна станів в процесі роботи системи (в залежності від часу). Для моделювання поведінки системи використовують:

- автомати - описують поведінку у термінах послідовності станів, через які проходить об'єкт протягом свого життя;
- взаємодії - описують поведінку у термінах обміну повідомленнями між об'єктами.

Статичні і динамічні моделі описують логічну організацію системи, відображають логічний світ програмного забезпечення. Моделі реалізації

забезпечують подання системи у фізичному світі, розглядаючи питання упаковки логічних елементів у компоненти та розміщення компонентів в апаратних вузлах. Засобом для побудови моделі реалізації є діаграми компонентів і діаграми розміщення.

3.3 Розробка структури програмного забезпечення при об'єктному підході

Діаграма пакетів

Один з найстаріших питань методології розробки програмного забезпечення: як розбити велику систему на невеликі підсистеми. Ми ставимо це питання, оскільки чим більше стають системи, тим важче розбиратися в них і під внесених до них зміни.

Структурні методи використовували функціональну декомпозицію, згідно якої вся система в цілому представляється як одна функція і розбивається на підфункції, які в свою чергу теж розбиваються на підфункції і т. д. Ці функції схожі на варіанти використання об'єктно-орієнтованої системи в тому, що функції являють собою дії, що виконуються системою в цілому.

У колишні часи процес і дані розглядалися окремо один від одного. Іншими словами, крім функціональної декомпозиції існувала також структура даних. Вона мала другорядне значення, хоча деякі методи інформаційних технологій групували запису даних у предметні області і формували матриці, щоб показати взаємозв'язок функцій і записів даних.

Саме з цієї точки зору ми можемо оцінити ті величезні зміни, які зробили об'єкти. Поділ процесу і даних залишилося в минулому, функціональна декомпозиція теж, однак найстаріше питання -, як і раніше все ще існує.

Одна з ідей полягає в угрупованні класів в блоки більш високого рівня. У мові UML такий механізм угруповання отримав назву пакет (package).

Множину ресурсів та описів класів називають пакетом.

На діаграмі видно складові частини ПЗ та їх взаємозв'язок.

Взаємозв'язок фіксується лише за умови, що зміни одного пакета впливають на інший. Він визначається взаємозв'язками класів та ресурсів, що перебувають в пакеті. Існують різні типи залежності:

- об'єкти одного класу посилають повідомлення об'єктів іншого класу;
- об'єкти одного класу звертаються до компонентів об'єктів іншого;
- об'єкти одного класу використовують об'єкти іншого у списку параметрів методів і т. п.

Найкращим варіантом можна виділити таку структуру, в якій опис та взаємодія всіх пакетів виокремлюється через уніфікований інтерфейс. Таким чином досягається такий розвиток подій, пакети не чіпають один одного під час зміни реалізації. Пакети, що пов'язані з усіма іншими в системі є глобальними.

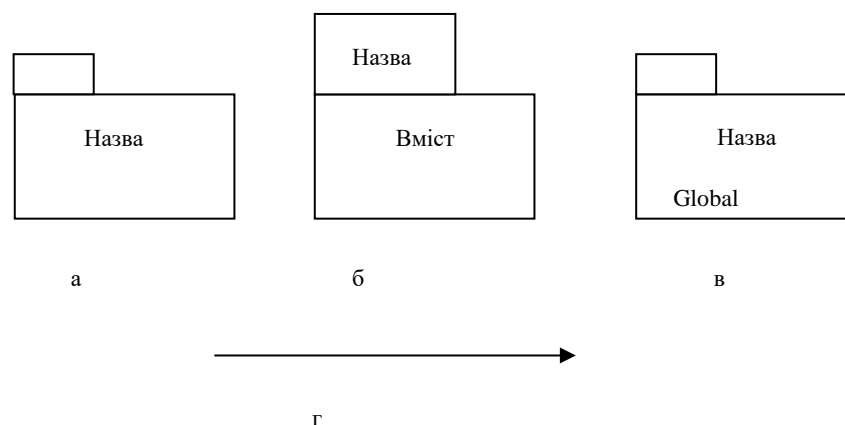


Рисунок 3.2 - Умовні позначення, застосовувані на діаграмах пакетів: а - пакет; б - пакет з позначенням вмісту; в - глобальний пакет; г - залежність класів (стрілка вказує напрям викликів).

На рис.3.3 показаний приклад діаграми пакетів системи рішення комбінаторно-оптимізаційних задач. Були виділені наступні групи класів або пакети: користувальницький інтерфейс - класи, які реалізують об'єкти інтерфейсу з користувачем; бібліотека інтерфейсних компонентів - класи, які реалізують інтерфейсні компоненти: вікна, кнопки, мітки; об'єкти управління - класи, які реалізують сценарії варіантів використання; об'єкти завдання - класи, які

реалізують об'єкти предметної області системи; інтерфейс бази даних - класи, які реалізують інтерфейс з базою даних; база даних; базові структури даних - класи, які реалізують внутрішні структури даних, такі, як дерева, n-зв'язні списки; обробка помилок - класи винятків, які реалізують обробку нештатних ситуацій.

Останні два пакети оголошені глобальними, так як їх елементи можуть використовувати класи всіх пакетів.

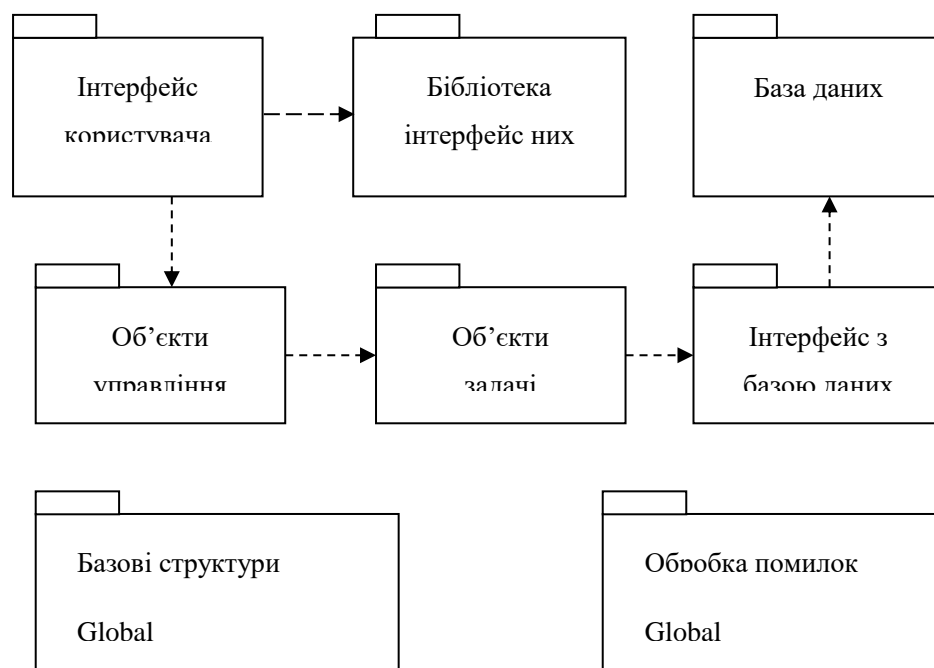


Рисунок 3.3 - Діаграма пакетів системи рішення комбінаторно-оптимізаційних задач

Діаграми класів

Діаграма класів - центральна ланка об'єктно-орієнтованих методів розробки програмного забезпечення (ПЗ), тому всі існуючі методи використовують діаграми класів в одній з відомих нотацій. Однак в основному діаграми класів в цих методах застосовують на етапі проектування, для того щоб показати особливості побудови конкретних класів. На відміну від раніше існуючих

нотацій, UML пропонує використовувати три рівня діаграми класів залежно від ступеня їх деталізації:

- концептуальний рівень, на якому діаграми класів, звані у цьому випадку контекстними, демонструють зв'язку між основними поняттями предметної області;
- рівень специфікацій, на якому діаграми класів відображають інтерфейси класів предметної області, тобто зв'язку об'єктів цих класів.
- рівень реалізації, на якому діаграми класів безпосередньо показують поля і операції конкретних класів.

На практиці ці моделі різні, а зв'язки між ними не такі однозначні. Визначення концептуальною моделлю понять предметної області не запорука того, що для реалізації буде використаний окремий клас. Однак у всіх трьох моделях нас цікавлять типи об'єктів (класи) та їх статичні відносини, що дозволяє використовувати єдину нотацію.

Кожну з перерахованих моделей використовують на конкретному етапі розробки:

- при аналізі - концептуальну;
- при проектуванні – рівня специфікації;
- при реалізації – рівня реалізації.

Діаграма класів описує типи об'єктів системи і різного роду статичні відносини, які існують між ними. Є два основних види статичних відносин:

- асоціації (наприклад, клієнт може взяти напрокат ряд відеокaset);
- підтипи (медсестра є різновидом особистості).

На діаграмах класів зображуються також атрибути класів, операції класів і обмеження, які накладаються на зв'язку між об'єктами.

Графічно клас в нотації мови UML зображується у вигляді прямокутника, який додатково може бути розділений горизонтальними лініями на розділи або секції. У цих секціях можуть зазначатися ім'я класу, атрибути та операції класу.

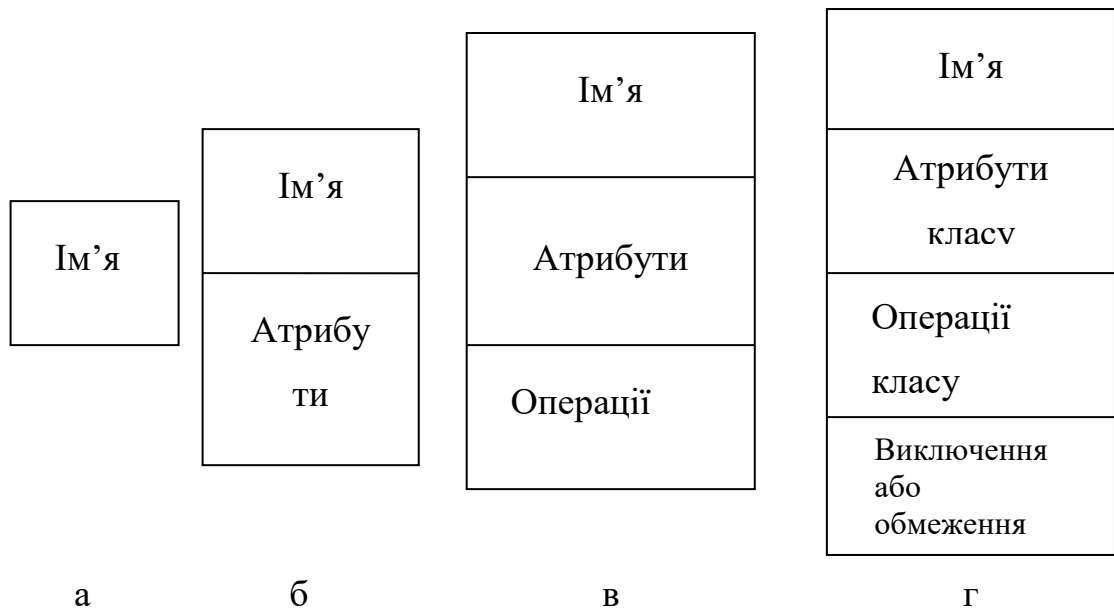


Рисунок 3.4 - Варіанти графічного зображення класу на діаграмі класів

На початкових етапах розробки діаграми окремі класи можуть позначатися простим прямокутником, в якому повинно бути вказано ім'я відповідного класу (рис. 3.4, а). По мірі опрацювання окремих компонентів діаграми опис класів доповнюється атрибутами (рис. 3.4, б) та операціями (рис. 3.4, в). Четверта секція (рис. 3.4, г) не обов'язкова і служить для розміщення додаткової інформації довідкового характеру, наприклад, про виключення або обмеження класу, відомості про розробника або мовою реалізації. Передбачається, що остаточний варіант діаграми містить найбільш повний опис класів, які складаються з трьох або чотирьох секцій.

3.4 Канали управління

Поняття каналу управління

Канал управління – послідовність дій та подій, які відбуваються у відповідь на появу деякої незапрошеної події, коли система перебуває у визначеному стані

(рис.3.5). Канал управління – важлива для аналітика концептуальна сутність, яка описує, що відбувається в системі, якщо незапрошена подія відбувається, коли вона знаходиться у визначеному стані.

Канал управління може включати в себе діяльність, яка виходить за рамки системи: в будь-якому місці каналу подія може породжуватися для термінатора, вимагаючи його виконувати деяку зовнішню діяльність. Якщо термінатор реагує на запрошену подію, то ця подія, а також зовнішню діяльність, вважають частиною каналу управління.

Якщо дія вздовж каналу управління породжує більш, чим одну подію, канал управління розгалужується так, що дві або більше гілки того самого каналу управління активні в один і той же час.

Кожна гілка каналу управління в кінцевому рахунку завершується. Це може відбутися одним з трьох способів.

Досягнута дія, яка не породжує ніяких подій, тоді гілка завершується цією дією.

Досягнута дія, яка породжує подію для термінатора, і він не реагує на запрошену подію. В цьому випадку діяльність завершується поза межами системи.

Досягнута дія, яка породжує подію для звільнення управляючих ресурсів. Тоді гілка завершується цією дією. Наступна діяльність тепер може відбуватися по іншому каналу управління, що затримувалося очікуванням доступності ресурсів, що управляються.

Наприклад, розглянемо проблему Абонент-Оператор (рис.3.5). Допустимо, що існує лише один оператор, який зайнятий в цій момент обслуговуванням абонента 9, і що Визначник Обслуговування знаходиться в стані 1: Очікування Абонента. Тоді, якщо настає незапрошена подія КТ1: Являється Абонент (абонент 10), можуть бути виконані тільки наступні дії:

Стан 1 Абонента: Очікування Оператора;

Стан 2 Визначник Обслуговування: Очікування Оператора.

Канал управління, який відходить від незапрошеної події, тимчасово зупинений, поки оператор не обслужить попереднього абонента, що породжує О-А2. Як показано на (рис.3.5), створення О-А2 є частиною каналу управління, який починається появою абонента 9. В той же час вважають, що перехід кінцевого автомату Визначника Обслуговування зі стану 2: Очікування Оператора до стану 3: Призначення є частиною каналу управління, що має справу з обслуговуванням абонента 10.

Схема каналу управління

На рис.3.5. приводиться приклад схеми каналу управління: графічне представлення послідовності подій і станів, які займають екземпляри, що приймають участь в деякому каналі управління. Кожний екземпляр з'являється окремо як строка станів, яку він займає в той час, поки канал еволюціонує. Стани, які займаються окремими екземплярами, з'єднуються стрілками, кожна з яких позначається подією, що визиває перехід до іншого стану. Якщо екземпляр породжує подію для іншого кінцевого автомату і ця подія визиває перехід, стрілка малюється від стану, яке породило подію, до переходу приймаючого екземпляра. Діаграма планується вздовж відносної вісі часу зі станами, які розміщуються на діаграмі в тому порядку (зверху до низу), в якому вони досягаються.

Час і канал управління

Схема каналу управління може використовуватися для аналізу часу, який потрібен системі, щоб відреагувати на незапрошену подію. Визначимо спочатку деякі терміни.

Час, на протязі якого екземпляр займає стан, який складається з часу дії и часу затримки.

Час дії – це час, який необхідний для виконання дії.

Час затримки- це час, на протязі якого екземпляр залишається в тому стані після завершення дії.

Час затримки іноді може бути повністю визначений екземпляром і станом: необхідно 10 хвилин на обслуговування Абонента 6 і 6 хвилин, щоб обслужити

Абонента 10. В якості альтернативи час затримки може визначатися взаємодіями між кінцевими автоматами: кількість часу, який необхідний Абоненту 10 для перебування в стані Очікування Оператора, - нуль, але фактичний час затримки для цього стану може бути значно більше, в залежності від зайнятості оператора.

Для опису часу каналів управління на схемі

Анотують кожний стан його часом дії;

Якщо час затримки стану визначається виключно екземпляром і станом, анотують перехід з цього стану з відповідним часом затримки;

Якщо час затримки стану визначається взаємодією між кінцевими автоматами, символ конденсатора розміщують на переході з того стану.

Час, який потрібен для реагування на незапрошену подію КТ1 (поява Абонента 10) тепер може бути обраховано зі схеми. Так як цей канал управління розділюється на дві гілки, збираємо дані в трьох частинах.

Загальний сегмент

Ця частина каналу управління починається з початком КТ1 і закінчується Визначником, який виконує дію стану Визначення. Допустимо, що Абонент 10 з'явився, коли оператор обслуговував Абонента 9 на протязі 7 хвилин. Тоді час, необхідний для загального сегменту:

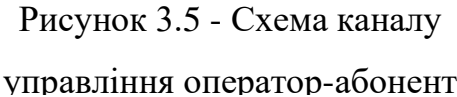


Рисунок 3.5 - Схема каналу управління оператор-абонент

Кінцевий автомат	Дія	Часова компонента	Кількість часу
Абонент 10	Очікує Оператора	Час дії	0:01
Абонент	Обслуговування	Час затримки	3:00
Абонент	Завершення обслуговування	Час дії	0:06
Абонент	Завершення обслуговування	Час затримки	0:00
Абонент	Очікування	Час дії	0:01
Визначник	Очікує оператора	Час затримки	0:01
Визначник	Визначення	Час дії	0:21
Всього			3:21

Ліва гілка. Ліва гілка цього каналу управління починається, коли Абонент 10 виконує дію стану Обслуговування, і закінчується, коли він завершує дію Задовільнений. Час, який потрібний для цієї гілки:

Абонент 10	Обслуговується	Час дії	0:01
Абонент 10	Обслуговується	Час затримки	6:00
Абонент 10	Задовільнений	Час дії	0:04
Всього			6:05

Права гілка. Права гілка починається, коли оператор виконує дію стану обслуговування і закінчується, коли він завершує дію Очікування. Час, потрібний для цієї гілки:

Оператор	Обслуговування	Час дії	0:01
Оператор	Обслуговування	Час затримки	6:00
Оператор	Закінчення обслуговування	Час дії	0:06

Оператор	Закінчення обслуговування	Час затримки	0:00
Оператор	Очікування	Час дії	0:01
Всього			6:08

Тоді загальний час реагування для незапрошеної події КТ1 – 9:26 для відповіді по лівій гілці та 9:29 – для відповіді по правій.

Схема каналу управління найбільш корисна як інструмент для розуміння взаємодії між найбільшою кількістю екземплярів. Однак, якщо кількість екземплярів збільшується, поняття розташування схеми відбувається дуже важко. Крім того, схема стає не контролюємо великою. Альтернативним підходом для розуміння каналу управління є імітування: метод, який виконує взаємодію кінцевого автомату, скоріш, у часі, чим у просторі, як показано на цій схемі.

Імітування

Щоб визначити, яким чином система як одне ціле відповідно реагує на незапрошену подію, можливо відстежувати канал управління за допомогою імітування. Потрібна стратегія, яка складається з трьох кроків:

1. Встановити початковий стан системи.
2. Прийняти незапрошену зовнішню подію і виконати канал управління, щоб побачити, що відбувається.
3. Оцінити кінцевий результат, якщо він коректний.

Встановлення стану системи

Завершений стан системи задається значеннями усіх атрибутів всіх екземплярів в системі. З метою розгляду будь-якого каналу управління ми повинні встановити значення для тих атрибутів, які впливають на розвиток каналу. Такі атрибути бувають двох типів: атрибути поточного стану (визначають, яка дія виконується, коли подія направлена до кінцевого автомату) і визначаючі атрибути.

Логіка дій може бути такою, що події, які породжені цією дією, залежать від значень атрибутів, відмінних від атрибуту поточного стану. Такі атрибути відомі як визначні, оскільки вони визначають, як розвивається канал управління, коли він проходить через дію (рис.3.5).

За допомогою систематичної процедури потрібні величини атрибутів встановлюють наступним чином:

1. Використовуючи моделі взаємодії об'єктів та моделі станів, намічають канал управління для визначення того, які моделі станів необхідно розглядати і скільки треба буде різних екземплярів кожного об'єкту. Обирають ідентифікатори для потрібних екземплярів і встановлюють значення для їх атрибутів поточного стану.
2. Аналізують кожну модель станів, яка розглядається в каналі управління для знаходження усіх визначаючих атрибутів. Встановлюють значення для визначаючих атрибутів.
3. З'ясовують, яким чином отриманий кожний визначаючий атрибут. Якщо він обчислюється моделями станів, які розглядаються в каналі управління, встановлюють величини для усіх атрибутів, які приймають участь в обрахуванні.
4. Перевіряють, чи не протирічать одне одному обрані значення атрибутів.

Виконання імітування

Як тільки потрібні значення атрибутів встановлені, можна починати відстеження каналу управління за допомогою імітатора.

Результати імітування представляються:

- Кінцевим значенням усіх атрибутів потокового стану;
- Кінцевим значенням всіх визначаючих атрибутів (включаючи атрибути, які приймають доля в обчисленні визначаючих атрибутів);
- Послідовність подій, які були направлені до шкірного термінатора.

Оцінюємо результати й переконуємося, що всі значення атрибутів не протирічать один одному і створена система з бажаною відповідною реакцією як наслідок прийнятої первісної не запитуваної події.

Інтерпретація паралелізму

ООА допускає дві інтерпретації паралелізму. Одночасна інтерпретація означає, що дві речі можуть відбуватися в один і той же час саме так, як і в повсякденному житті.

Інтерпретація, що чергується, означає, що тільки одна річ може відбутися в будь-який момент. Для досягнення ефекту одночасного процесу два або більше процеси повинні бути розділені на під процеси, і ці процеси виконуються. Це точка зору на паралелізм, яка підтримується багатозадачними операційними системами.

Правила часу ООА

Формалізація в ООА дає певні твердження та припущення відносно часу, які надалі називають правилами часу. Більшість правил часу однакові як для одночасної, так і для інтерпретації часу, що чергується.

Правила відносності дій:

1. Тільки одна дія даного кінцевого автомату може виконуватися в будь-якій точці часу.

2. Дія потребує часу на виконання. В результаті цього ООА відрізняється від багатьох методів аналізу, які передбачають, що дії відбуваються миттєво.

3а. Паралелізм для одночасної інтерпретації: дії в різних кінцевих автоматах можуть виконуватися одночасно.

3б. Паралелізм для інтерпретації, що чергується: тільки одна дія може виконуватися в будь-який час. Якщо дія отримує управління, воно виконується до завершення без переривань. Дії в різноманітних кінцевих автоматах чергуються довільно. Іншими словами, дія є одиницею чергування.

4. Після ініціалізації, дія кінцевого автомату повинна закінчитися до того, коли він зможе прийняти іншу подію.

Правила несуперечливих даних

1. Коли дія завершується, вона може залишати систему несуперечливою або шляхом запису даних для створення несуперечливої картини, або породженням подій, які примушують кінцеві автомати прийти у відповідність зі зміною даних, створених відправником подій. Системі може знадобитися деякий час для того, щоб стати повністю несуперечливою при другій інтерпретації часу.

2. Обов'язки аналітика – гарантувати, що дані, необхідні дією, несуперечливі, або що дія передбачає суперечливість, яка обумовлена часом розповсюдження.

Правила подій

1. Подія ніколи не пропадає: кожна подія досягає кінцевого автомату або термінатору, до якого було направлена.

2. Подія використовується, коли приймається кінцевим автоматом, потім пропадає як подія і не може бути повторно використано. Це правило аналогічно більшості механізмів задач взаємодії, в яких повідомлення не може бути отримано ще раз. Як тільки повідомлення прийнято, воно зникає як повідомлення.

3. Коли подія породжується, воно одразу стає доступним для кінцевого автомату або термінатора, до якого направлено.

4. Коли екземпляр завершує дію, то вважається, що він тепер знаходиться в новому стані. Тоді

- екземпляр потребує доступну подію, яка направлена до нього, якщо взагалі такі події існують в цей час (однократна інтерпретація);

- в будь-який невизначений час після завершення дій кінцевий автомат набере нову доступну подію, якщо взагалі таке існує (чергова інтерпретація).

5. Багатократні події можуть виступати в якості чекаючих обробки кінцевим автоматом.

При одночасній інтерпретації часу декілька кінцевих автоматів змогли би породжувати події для деякого приймача на протязі того періоду, коли приймач

буде зайнятий виконанням дії. Це змусило б події очікувати приймач, який завершує дію, до того моменту, коли може бути прийнято інша подія.

При черговій інтерпретації часу декілька кінцевих автоматів можуть породжувати події для визначеного приймача, перш ніж приймаючому кінцевому автомату дозволять отримувати події і виконувати дію (тому що дії чергуються довільно). Це приводить до того, що багатократні події знову будуть чекати обробки одним кінцевим автоматом.

6. Якщо кінцевий автомат породжує багатократні події для одного приймаючого екземпляру, події будуть отримані в порядку, що згенерував.

Якщо є події, які чекають обробки для деякого кінцевого автомату, які були породжені різними відправниками, то не визначено, яка подія буде прийнята з початку.

3.5 Моделі процесів

Усі процеси, які відбуваються в системі, встановлюються в діях. Тепер більш детально розглянемо процеси, які складають дію.

Раніше розглядалася логіка кожної моделі станів і системи, як одного цілого, коли різні дії виконуються по відношенню один до одного, а не точні подробиці процесів всередині дій.

Тепер розглянемо алгоритмічну або функціональну природу дій. Мета полягає в тому, щоб розділити кожну дію на фундаментальні процеси, які взяті разом, визначають потрібне функціональний склад системи.

Основний інструмент, який використовується для розподілу, - діаграма потоків даних дій, графічне представлення внутрішньої організації дій. Спочатку представимо діаграму потоків даних дій з перспективою нотації. Потім розглянемо правила і керівні принципи для окремих дій в процесах, методики визначення деталей процесу, а також деякі робочі продукти, корисні в управлінні цим аспектом аналізу.

Діаграми потоків даних дій (ДПДД) забезпечують графічне представлення модулів процесу в радіусі дії і взаємодії між ними. Діаграма потоків даних дій ґрунтується на нотації, яка була введена наприкінці 70-х років Йорденом, Константином та де Марко, а пізніше розширена іншими [3,4,5] з метою включення понять, що відносяться до подій.

Діаграма потоків даних дій ООА повністю подібна діаграмам, які представлені де Марко, за виключенням деяких важливих моментів:

- документ «Опис об'єктів та атрибутів» використовується замість традиційного впорядкованого в алфавітному порядку словника даних;
- в ООА задача розбивається спочатку на об'єкти, потім на дії, потім, на процеси всередині дії, видаючи в нижній частині цієї ієрархії одну одну образну діаграму потоків даних для кожної дії. Це відрізняється від традиційних методик, які описують послідовну декомпозицію функції і проводять багаторівневий набір діаграм потоків даних;
- управління представляється на діаграмах потоків даних дій: явно через потоки управління чи неявно через визначені потоки даних;
- умовні висновки представляються явно.

Процеси і потоки даних

На діаграмі потоків даних дій представлені викладки, які необхідні цим діям, у вигляді його основних модулів, що називаються процесами. Кожний процес представляється овалом, що анотується як ідентифікатором процесу, так і значимим ім'ям, які описують мету або функцію процесу.

Більшість процесів потребують вхідних даних, для того щоб виконувати функції, і, як результат, виробляють вихідні дані. Якщо процес потребує вхідних даних, ці дані показані як потік даних, який направлений від процесу.

Стійкі дані

Дані, які продовжують існувати після того, як дія закінчена, відомі як стійкі. Стійкі дані представляються на ДПДД як архів даних, логічний еквівалент

таблиці в базі даних, файлі або наборі системних змінних. Якщо архів даних забезпечує дані для (або отримує дані з) процесу, архів даних і процес зв'язується потоком даних.

ООА використовує три різних типу архіву даних: об'єктний архів даних; архів даних поточного часу; архів даних таймеру.

Об'єктний архів даних

Дані, які описують усі екземпляри об'єкту, представлені як архів даних, що відмічений ім'ям об'єкту. Об'єктний архів даних може інтерпретуватися як таблиця, що має значення усіх атрибутів усіх екземплярів об'єкту.

Альтернативний об'єктний архів даних може розглядатися як дані екземпляра для усіх екземплярів класу в ООД.

Якщо процес прочитує атрибути з (або записує атрибути) об'єктного архіву даних, потік даних, який пов'язує процес та об'єктний архів даних, маркується прочитаними або записуваними атрибутами.

Архів даних поточного часу. Дані, що описують поточний час, представлені на ДПДД як архів даних, який позначений словосполученням «Поточний час». Цей архів даних може інтерпретуватися як системні часи. Архів даних часу – джерело будь-яких даних поточного часу, необхідних процесу.

Потоки даних, що зв'язують архів даних поточного часу з процесом, маркуються відповідно названими змінними, які представляють поточний час: день, місяць, годину і т.п., - чи просто поточний час. Для того, щоб отримати задовільне планування ДПДД, буд-який архів даних може повторюватися стільки разів, скільки необхідно.

Отримані події

Події, які приймаються моделлю станів і таким чином визивають ініціалізацію дії, відображується на ДПДД як один чи більше потік даних події – потік даних події визначається іменами атрибутів, які переносяться подією і потребується

процесом. Так як в загальному випадку дія може бути ініціалізовано будь-яким з декількох різних подій, мітки подій не показуються з отриманими даними події.

Породжувані події

Якщо процес породжує події, ця подія показується як потік даних, направлених від процесу. Потік даних визначається міткою, значенням та даними події так, як на діаграмі переходів в стан або моделі взаємодії об'єктів.

Ідентифікатори процесів

Кожний процес на ДПДД визначений для об'єкту з інформаційної моделі у відповідності з керівними принципами. Об'єкт, для якого визначений процес, відображується в ідентифікаторі процесу, який має «вид», «вільний номер процесу», «Об'єкт» може визначатися або номером об'єкту, або ключовим літералом, призначеним об'єкту на інформаційній моделі.

Потоки даних між процесами

Якщо дані створюються одним процесом і використовуються іншими, потік даних рисується між двома зв'язаними процесами. Потік даних повинен бути помічений ім'ям атрибуту, який переноситься потоком даних. Загальне для атрибутів на такому потоці даних – це необхідність мати два ім'я: одне – що базується на перспективі процесу, який створює потік даних, а інше – базується на перспективі процесу, який використовує його. Для економії місця на ДПДД частина повного ім'я атрибуту «ім'я об'єкту» іноді опускають, якщо вона може бути отримана за допомогою наступного правила.

Правило. *Вважається, що будь-який атрибут, створений або той що використовується процесом, є атрибутом об'єктом, для якого даний процес визначений, якщо не стверджується навпаки.*

Таймери

Для підтримки таймерних узгоджень, п'ять процесів визначені як частина формалізації ООА. Ці процеси (рис.3.7) використовуються на буд-якій ДПДД по мірі необхідності.

Потік управління є графічним представленням обмеження на порядок виконання процесу. ДПДД використовує два потоки управління: безумовний та умовний.

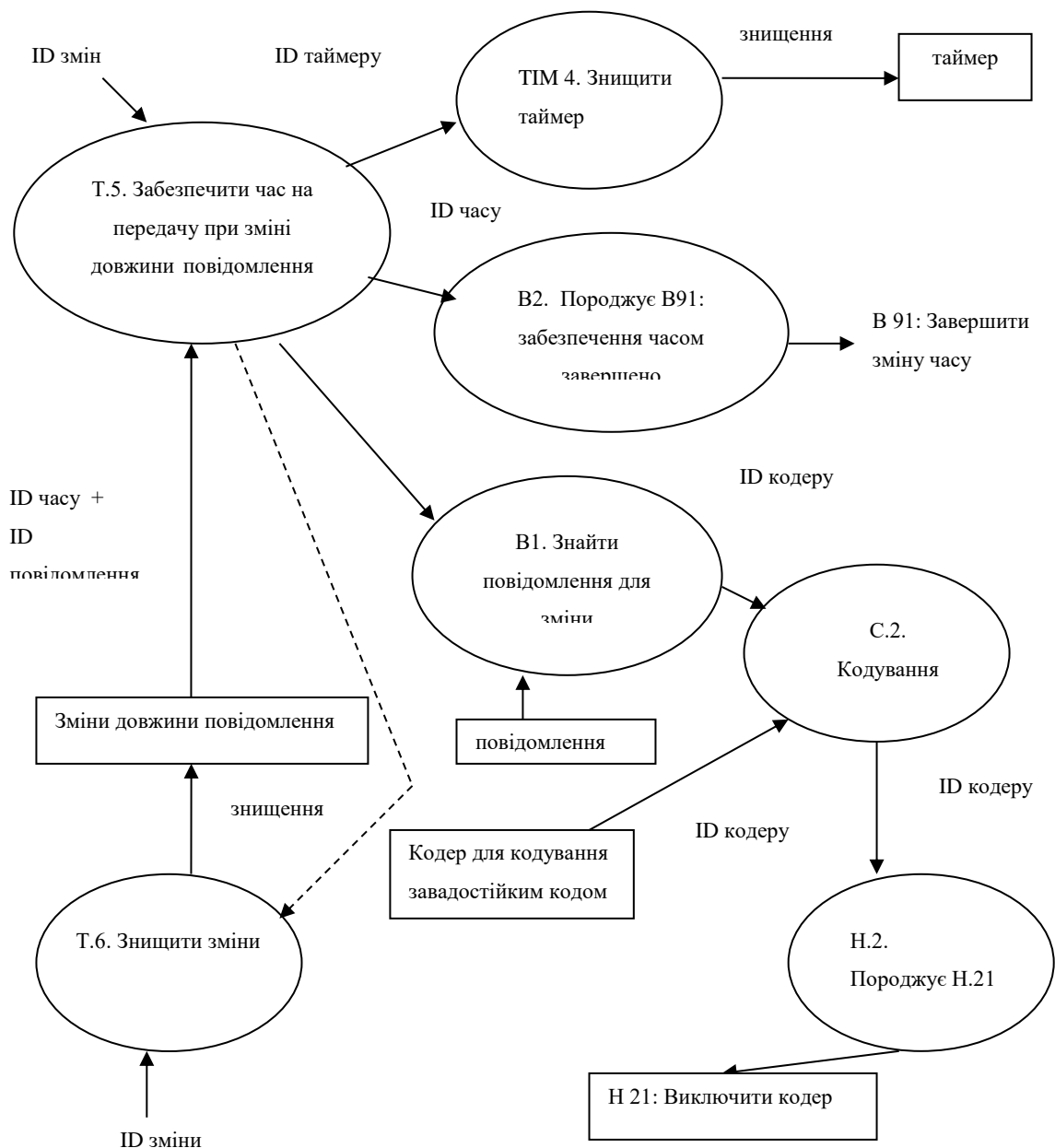


Рисунок 3.6- ДПДД для стану забезпечення часу на передачу при зміні довжини повідомлень

Безумовні потоки управління

На рис.3.6 безумовний потік управління (відображений як пунктирна лінія) виконаний від процесу TR.5 до процесу TR.6 для вказівки того, що TR.5 повинен виконуватися до TR.6. Це необхідно, щоб гарантувати, що TR.5 зможе відновити зміни довжини повідомлення, ID часу та ID таймеру, до того, як TR.6 знищить екземпляр.

Умовні потоки управління

Крім обмежень на порядок виконання процесів, умовний потік управління забезпечує графічне представлення умов, при яких процес виконується (або не виконується).

Нестійкі дані

Процес може створювати дані, які призначені тільки для використання іншим процесом, тобто дані, які завершують своє існування, якщо дія завершена. Такі дані називаються нестійкими. Якщо потік даних несе нестійкі дані, потік даних, який пов'язує створюючий та використовуючий процеси, визначається відповідним ім'ям та анотується, для того щоб вказати, що він переносить нестійкий (тобто непостійний) елемент даних.

Багатократні екземпляри

Якщо потік даних може переносити атрибути для більш чим одного екземпляру об'єкту, потік даних відмічається іменами атрибутів, які записуються у фігурних дужках.

Умовні потоки даних

Якщо процес створює дані тільки при визначених умовах, вихідні дані показані як умовний потік даних – регулярний потік даних, відмічених поперечною рисою.

3.6 Об'єктно-орієнтована модель системи управління мережею

Давайте будемо намагатись реалізувати ОО модель використовуючи ООА.

Для ООА справедлива таблиця станів Мура, що включає в себе:

1. Різноманіття станів. Кожен з них являється стадією життєвого циклу екземпляру;
 2. Різноманіття подій. Кожна з них описує розвиток:
 - переходи. Описують, що коїться з об'єктом після події;
 - дія. Описано необхідний порядок дій, тригером для яких є досягнення стану.
- Тут є правило – одна дія-один стан.

Стан можна представити як діаграму переходу в стан. Положення об'єктів з набором відповідних правил та законів називають станом. На діаграмі їх показують рамками. Опис необхідний для кожної дії.

У такій моделі об'єкти володіють динамікою. Давайте їх розглянемо. ВПУ та ОУ являються статичними, бо інформація надається у автоматичному режимі, ВПУ та ОУ тільки зберігають так контролюють всі ці дані. Можна сказати, що такі об'єкти беруть на себе роль ревізора, що має змогу отримати будь-яку потрібну операцію в будь-яку одиницю часу.

Побудуємо моделі станів основних об'єктів системи управління.

Модель станів з'єднувальної лінії

Динаміка переходів з'єднувальної лінії із стану в стан показано на рис.3.8. Як видно з рисунка ЗЛ може знаходитись в чотирьох станах:

- 1.Готова і не діє; 2.Прийом повідомлення; 3.Передача повідомлення; 4.Аварія.

Переходи вказані стрілками з міткою події які ініціюють перехід(стрілка спрямована від поточного стану до наступного).На рис.2.16 показана діаграма переходу в стани для з'єднувальної лінії.

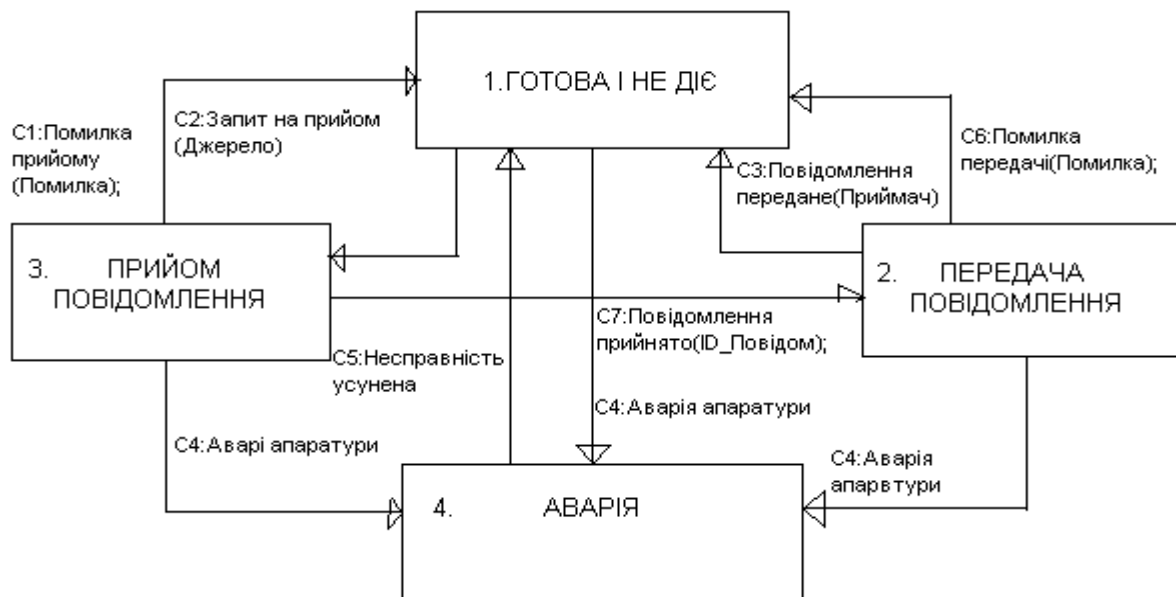


Рисунок 3.7 - ДПС для з'єднувальної лінії.

Ця діаграма може бути формалізована таблицею переходу в стани (ТПС) (таблиця 3.2). В ТПС кожний рядок являє один із можливих станів молі станів і кожний стовпчик – подію, яка має цю модель станів як призначення. Комірки заповнюються для визначення того, що відбувається, коли екземпляр в цьому стані (рядок) приймає конкретну подію (стовпчик).

Комірки можуть прийняти наступні значення:

1. Новий стан – комірка заповнюється іменем нового стану, у яке переходить екземпляр, коли він в стані, визначеному рядком, приймає подію, визначену стовпчиком. Новий стан точно відповідає стрільці переходу на ДПС. Подія може примушувати об'єкт переходити в той же стан, в якому він не знаходиться і примушує об'єкт повторно виконати дію зв'язану з цим переходом.

2. Подія ігнорується – об'єкт відмовляється реагувати на дану подію, коли він знаходиться у визначеному стані. Коли подія ігнорується, об'єкт залишається у колишньому стані і не виконує повторну дію.

3. Не може виникнути – подія не може виникнути, коли об’єкт знаходиться у визначеному стані. Значення «Не може виникнути» заноситься в тому випадку, якщо подія не може виникнути у реальному світі.

	C1	C2	C3	C4	C5	C6	C7
1.Готовий і не діє	I	3	I	4	Н	Н	I
2.Передача повідомлення	Н	I	1	4	Н	1	Н
3.Прийом повідомлення	1	I	Н	4	Н	Н	2
4.Аварія	Н	Н	Н	Н	1	Н	Н

Таблиця 3.2

«Н» - не може відбутися,

«I» - ігнорується.

Модель стані таймеру з'єднувальної лінії

Атрибути таймеру (ТМ) являються:

- ID Таймеру – ідентифікатор власне таймеру;
- Залишений час – час, який повинен пройти до того моменту, коли таймер видає сигнал;

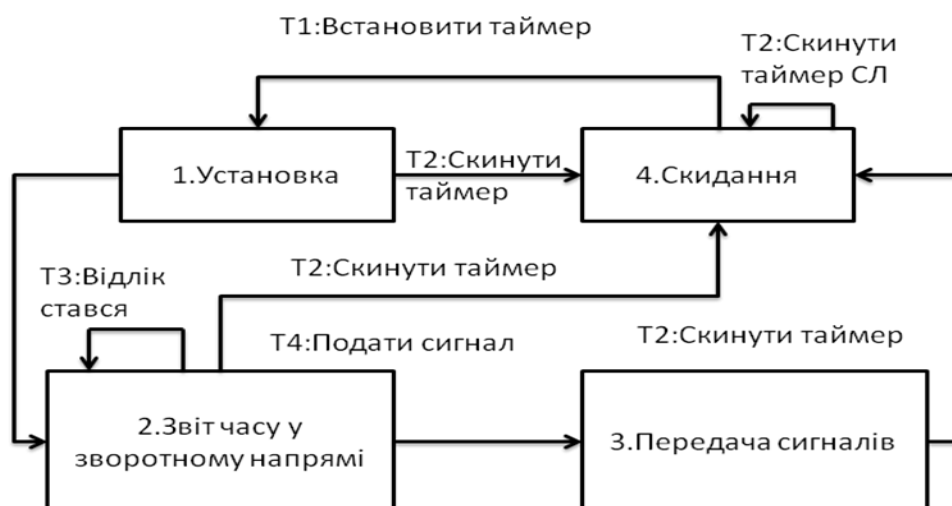


Рисунок 3.8 - Модель стану Таймера 3Л

	T1:Встановити таймер	T2:Скинути таймер	T3:Відлік стався	T4:Подати сигнал
1.Установка	Н	4	2	І
2.Відлік часу в зворотному напрямленні	І	4	Н	3
3.Подача сигналів	І	4	І	4
4.Скидання	І	Н	І	Н

Таблиця 3.3

-ID екземпляра – ідентифікатор, який передається з породженням події.

Модель станів Таймера з'єднувальної лінії представлена на рис.3.9.Модель побудована на механізмі рахунка в зворотному напрямленні. ТПС для ЗЛ представлена в табл. 3.3.

Умовні позначення псевдокоду для Таймера ЗЛ :

Створення таймера записуємо:

- ID таймера – таймер ЗЛ;

Для встановлення таймера породжують подію :

- T1 : Встановити таймер.

Для Скидання таймера породжують подію :

- T2 : Скинути таймер.

Для читання інтервалу часу, що залишився на таймері записуємо:

- Ще залишилось = прочитати час таймера ЗЛ, що залишився.

Видалення таймера записуємо як :

- Видалити таймер ЗЛ.

Модель станів черги повідомлень

Досить динамічним об'єктом є Черга Повідомлень (MesageQueue) – це службовий об'єкт, він співпрацює зі CommChanel і своїм власником – ОУ або ВПУ, і відповідає за внутрішню обробку (зберігання і документування) інформації. Його ДПС показана на рис.3.9, а ТПС для черги повідомлення представлена в табл.3.4.



Рисунок 3.9 - ДПС для черги повідомлень.

	01:Запит на прийом повідомлення	02:Кінець прийому	03:Канал готовий (ID_Каналу)	04:Кінець передачі (ID_Приймача)
Очікування	2	I	3	I
Прийом	I	1	H	H
Передача повідомлення	I	H	I	1

Таблиця 3.4

Модель станів об'єкту управління

Останнім об'єктом в моделі, є об'єкт управління. Його динаміка ілюструється ДПС і ТПС (табл.3.5).

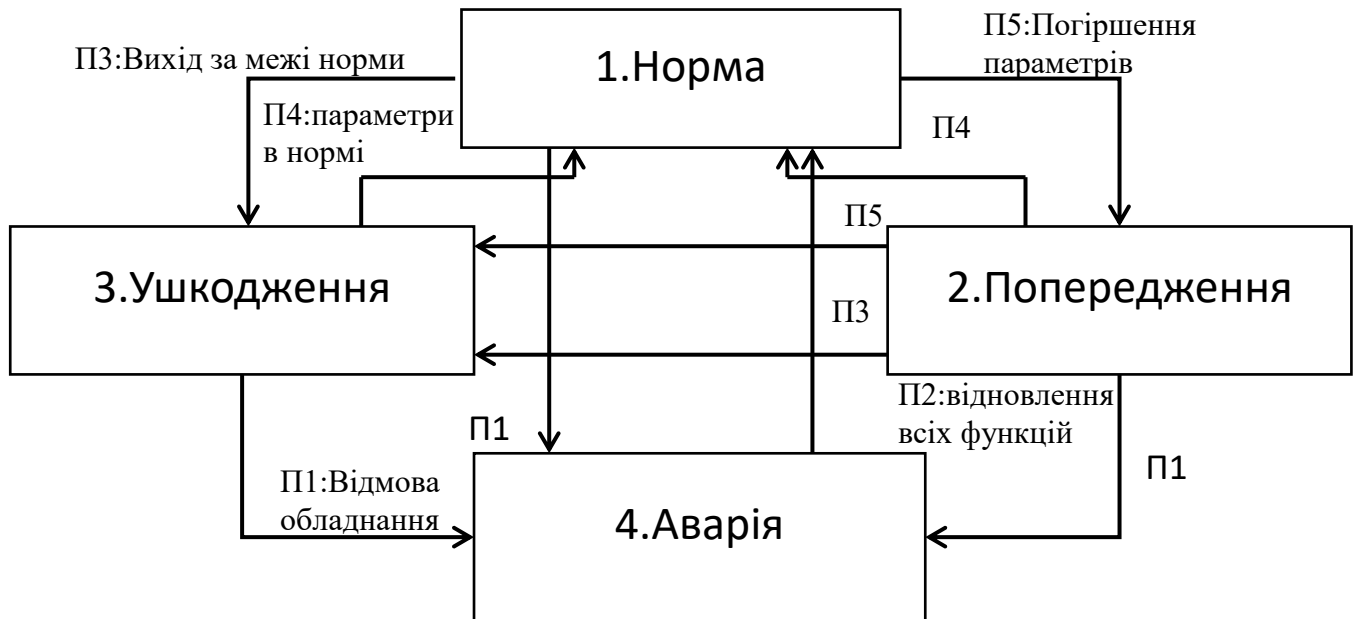


Рисунок 3.10 - ДПС для ОУ

	П1:Відмова апаратури	П2:Відновлення усіх функцій	П3:Вихід за межі норми	П4:Параметри в нормі	П5:Погіршення параметрів
Норма	4	I	3	I	2
Попередження	4	H	3	1	3
Пошкодження	4	H	H	1	H
Аварія	I	1	H	H	H

Таблиця 3.5

Дана модель об'єкту управління (ОУ) – це загальний випадок. Але вона закладає фундамент для подальшого розвитку. Ця модель описує динаміку в цілому, тобто будь-якому об'єкту управління, вузлу апаратури чи тракту властива описана поведінка, Але конкретний вид ОУ може знаходитися і в інших станах, властивих цьому виду ОУ.

3.7 Об'єктно-орієнтована модель системи управління в основних 4-х станах

Побудуємо об'єктно-орієнтовану модель системи управління телекомунікаційною мережею в 4-х основних станах.

	1	2	3	4	5	6	7	8	9	10
--	---	---	---	---	---	---	---	---	---	----

1 Прийом сигналів	2	Н	Н	3	І	Н	Н	Н	Н	І
2 Режим «норма»	І	5	Н	Н	Н	Н	Н	Н	Н	Н
3 Очікування	Н	Н	Н	І	1	І	6	Н	4	Н
4 Готовність прийому	Н	Н	Н	Н	Н	Н	Н	І	І	1
5 Формування сигналів	Н	І	6	Н	Н	Н	Н	Н	Н	Н
6 Передача	Н	Н	І	Н	Н	3	І	4	Н	Н

Таблиця 3.6 - Таблиця переходу в стани для режиму «норма»

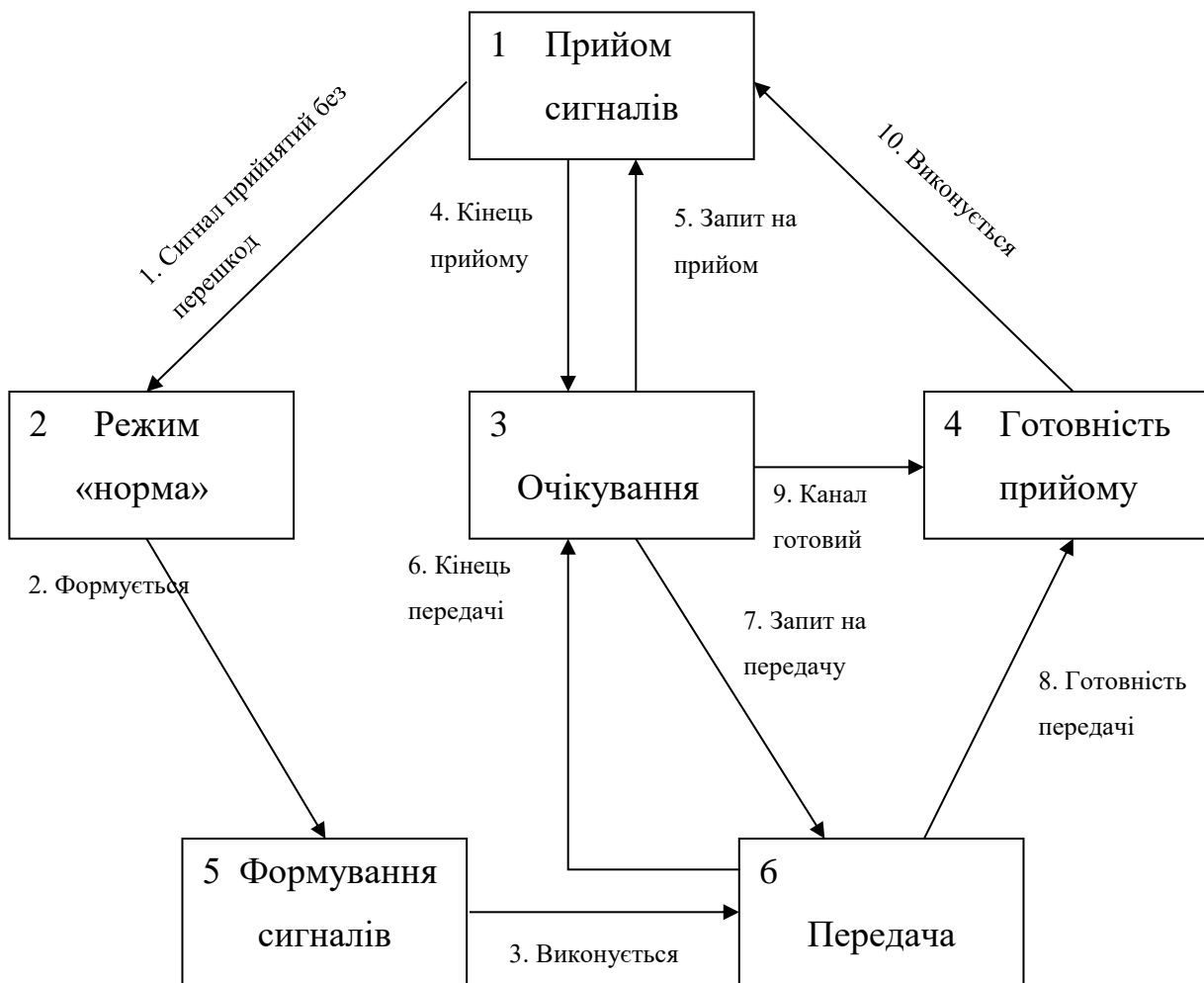


Рисунок 3.11 - Об'єктно-орієнтована модель системи управління телекомунікаційною мережею в режимі «норма» вузлового пункту управління, з'єднувальної лінії

Об'єктно-орієнтовану модель системи управління телекомунікаційною мережею в режимі «норма» вузлового пункту управління, з'єднувальної лінії зображено на рис. 3.11.

Переходи вказані стрілками з міткою події які ініціюють перехід (стрілка спрямована від поточного стану до наступного).

Принцип дії переходу від одного стану в інший показано в таблиці 3.6

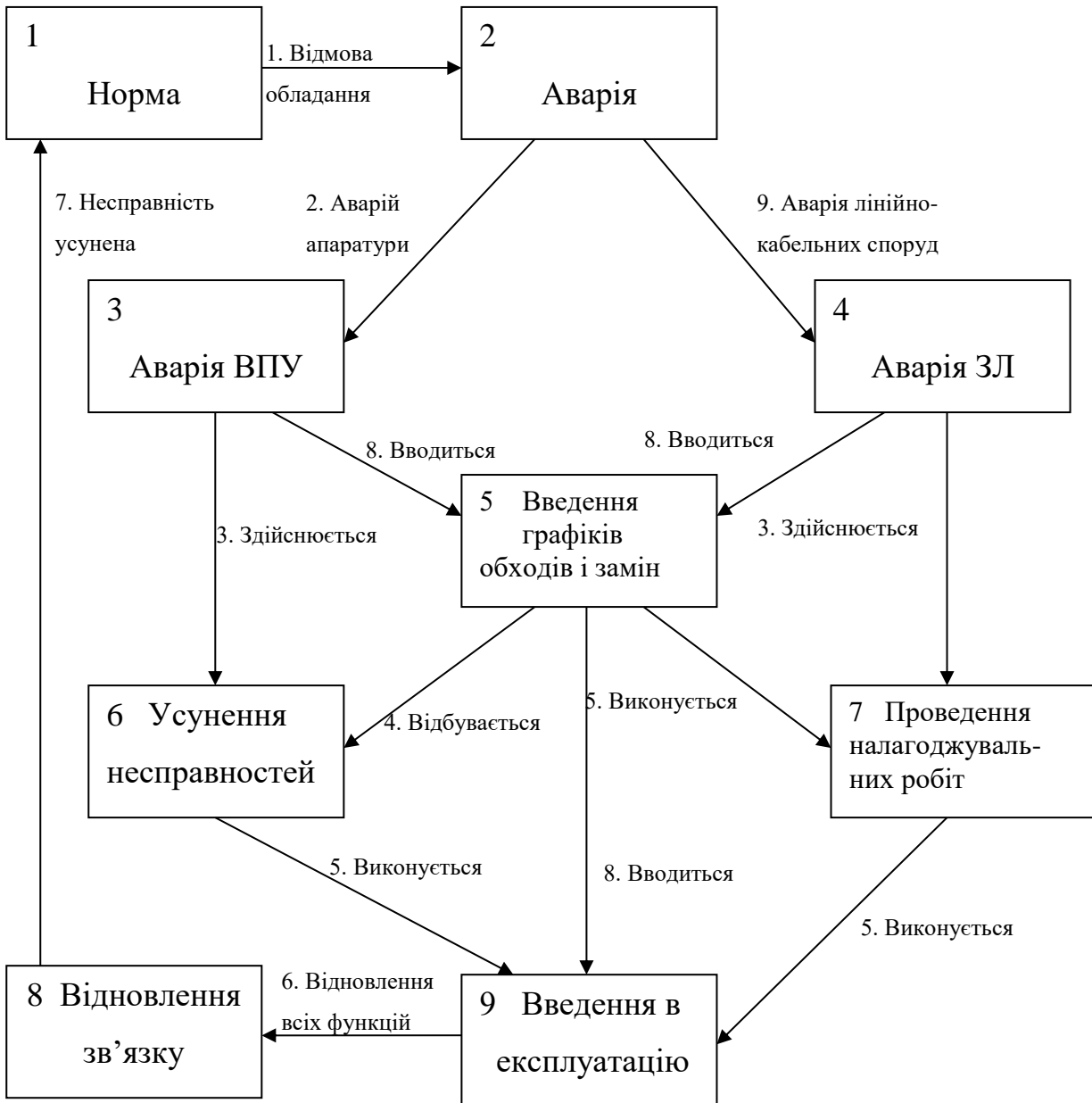


Рисунок 3.12 – Об'єктно-орієнтована модель системи управління телекомунікаційною мережею при аварії вузлового пункту управління, з'єднувальної лінії

Об'єктно-орієнтовану модель системи управління телекомунікаційною мережею при аварії вузлового пункту управління, з'єднувальної лінії представлено на рис. 3.12.

Побудуємо таблицю переходу в стани при аварії.

	1	2	3	4	5	6	7	8	9
1 Норма	2	Н	Н	Н	Н	Н	І	Н	Н
2 Аварія	І	3	Н	Н	Н	Н	Н	Н	4
3 Аварія ВПУ	Н	І	6	Н	Н	Н	Н	5	Н
4 Аварія ЗЛ	Н	Н	7	Н	Н	Н	Н	5	І
5 Введення графіків обходів і замін	Н	Н	Н	8	8	Н	Н	8	Н
6 Усунення несправностей	Н	Н	І	Н	8	Н	Н	Н	Н
7 Проведення налагоджувальних робіт	Н	Н	І	Н	8	Н	Н	Н	Н
8 Введення в експлуатацію	Н	Н	Н	І	І	9	Н	І	Н
9 Відновлення зв'язку	Н	Н	Н	Н	Н	І	1	Н	Н

Таблиця 3.7 – Таблиця переходу в стани при аварії

Об'єктно-орієнтовану модель системи управління телекомунікаційною мережею при ушкодженні вузлового пункту управління, з'єднувальної лінії представлено на рис. 3.13.

Побудуємо таблицю переходу в стани при ушкодженні.

	1	2	3	4	5	6	7	8
1 Норма	2	Н	Н	Н	Н	Н	Н	І
2 Ушкодження	І	3	4	Н	Н	Н	Н	Н
3 Ушкодження ВПУ	Н	І	Н	5	6	Н	Н	Н
4 Ушкодження ЗЛ	Н	Н	І	Н	6	5	Н	Н
5 Усунення несправностей	Н	Н	Н	І	Н	І	Н	1
6 Введення графіків обходів і замін	Н	Н	Н	Н	І	Н	7	Н
7 Відновлення зв'язку за рахунок заміни	Н	Н	Н	Н	Н	Н	І	1

Таблиця 3.8 – Таблиця переходу в стани при ушкодженні



Рисунок 3.13 – Об'єктно-орієнтована модель системи управління телекомунікаційною мережею при ушкодженні вузлового пункту управління, з'єднувальної лінії

Побудуємо таблицю переходу в стани при ушкодженні.

	1	2	3	4	5	6	7	8
1 Норма	2	Н	Н	Н	Н	Н	Н	І
2 Ушкодження	І	3	4	Н	Н	Н	Н	Н
3 Ушкодження ВПУ	Н	І	Н	5	6	Н	Н	Н
4 Ушкодження ЗЛ	Н	Н	І	Н	6	5	Н	Н
5 Усунення несправностей	Н	Н	Н	І	Н	І	Н	І
6 Введення графіків обходів і замін	Н	Н	Н	Н	І	Н	7	Н
7 Відновлення зв'язку за рахунок заміни	Н	Н	Н	Н	Н	Н	І	І

Таблиця 3.9. Таблиця переходу в стани при ушкодженні

Об'єктно-орієнтовану модель системи управління телекомунікаційною мережею при попереджувальному стані вузлового пункту управління, з'єднувальної лінії представлено на рис. 3.14.



Рисунок 3.14 – Об'єктно-орієнтована модель системи управління телекомунікаційною мережею при попереджувальному стані вузлового пункту управління, з'єднувальної лінії

На основі цих даних також були розроблені алгоритми в основних 4-х станах, які приведені в додатках:

- алгоритм роботи мережі в режимі «норма» вузлового пункту управління, з'єднувальної лінії
- алгоритм роботи мережі в режимі «аварія» вузлового пункту управління, з'єднувальної лінії;
- алгоритм роботи мережі в режимі «ушкодження» вузлового пункту управління, з'єднувальної лінії;
- алгоритм роботи мережі при попереджувальному стані вузлового пункту управління, з'єднувальної лінії.

	1	2	3	4	5	6	7	8	9
1 Норма	2	Н	4	Н	Н	Н	Н	5	3
2 Попередження	І	4	Н	Н	Н	Н	Н	Н	Н
3 Відновлення зв'язку	Н	Н	І	Н	Н	Н	Н	Н	І
4 Усунення попереджувального стану	Н	І	І	6	І	Н	Н	Н	Н
5 Введення в експлуатацію	Н	Н	3	Н	Н	Н	І	І	Н
6 Отримання дозволу	Н	Н	Н	І	4	7	Н	Н	Н
7 Введення графіків обходів і замін	Н	Н	Н	Н	Н	І	5	Н	Н

Таблиця 3.10. Таблиця переходу в стани при попереджувальному стані

Висновки до розділу

У цьому розділі було проведено побудова концептуальної моделі предметної області. Описано динамічні, статичні моделі та моделі реалізації об'єктно-орієнтованих програмних систем. За допомогою об'єктно-орієнтованого моделювання ми змодельовано об'єктно-орієнтовані моделі, які покривають весь спектр найважливіших конструкторських рішень, які необхідно розглядати при розробці складної системи. Також розглянуто поведінку системи в 4-х основних станах: норма, аварія, ушкодження, попередження

4. РОЗРОБКА СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

4.1.1 Уявлення про зміст ідеї

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробити та виготовити пристрій розширення функціоналу для систем управління телекомунікаційною мережею.	1. Підприємства із застарілим телекомунікаційним обладнанням.	Зменшення витрат на повне переоснащення апаратури, можливість відтягнути цей процес для того, щоб відповідно підготуватись.

Таблиця 4.1 - Опис ідеї стартап-проекту

4.1.2 Аналіз потенційних техніко-економічних переваг ідеї

№ п/п	Техніко-економічні характеристики ідеї	(потенційні)товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкурент 1	Конкурент 2	Конкурент 3			
1	Собівартість продукції у \$	2000	3750	6000	5000	-	-	+
2	Чистий прибуток	16000	15400	15500	15450	-	-	+

Таблиця 4.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

4.2 Технологічний аудит ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Пристрій розширення функціоналу	Технологія 1 Програмне забезпечення, модель, саме пристрій.	+	+
Обрана технологія реалізації ідеї проекту: 1				

Таблиця 4.3. Технологічна здійсненність ідеї проекту

Висновок: технологічна реалізація продукту – можлива, вибрана технологія №1.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

4.3.1 Аналіз попиту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	1
2	Загальний обсяг продажу, грн./ум.од	300000
3	Динаміка ринку	Стагнує
4	Наявність обмежень для входу	Серйозних обмежень немає, зазвичай підприємства обладнуються саморобними системами
5	Специфічні вимоги до стандартизації та сертифікації	Пристрій має бути сертифікованою та захищено патентом.
6	Середня норма рентабельності в галузі або по ринку, %	8%

Таблиця 4.4 - Попередня характеристика потенційного ринку

Беручи до уваги те, що конкуренція на ринку невисока, а також сам ринок стоїть на місці, вважаю, що стартап-продукт доцільний у даний час. Норма рентабельності достатня.

4.3.2 Аналіз груп клієнтів

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Економія ресурсів, усунення потреби заміни всього обладнання.	Підприємства, що займаються наданням телекомунікаційних послуг та використовують застаріле обладнання.	Відсутні.	- надійна, безотказна робота, легкість в експлуатації - надання гарантійного терміну, підтримки

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

4.3.3 Аналіз загроз та можливостей

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Поява аналогічного рішення.	Допрацювання та додання функціоналу разом зі зниженням вартості.
2	Ресурси	Недостатня кількість грошей для реалізації продукту	Залучення інвесторів. Можливо, отримання кредитних коштів.

Таблиця 4.6 - Фактори загроз

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Наявність продукту	Вихід на ринок	Постійне покращення та доробка продукту
2	Випуск аналогічного рішення	Надання унікального та вигідного користувачам продукту	Оптимізація параметру ціна-якість
3	Зворотній зв'язок	Отримання інформації щодо збоїв у роботі або проблем іншого характеру під час роботи	Виправлення негативних сторін продукту

Таблиця 4.7 - Фактори можливостей

4.3.4 Аналіз конкуренції

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції - чиста	Товари на ринку більш менш схожі між собою та майже не впливають один на одного	Внесення унікального функціонала, нижча вартість на ринку
2. За рівнем конкурентної боротьби - світовий	Товари не прив'язані до конкретної країни. Штат розробників міжнародний	Створення представників у країнах, які розглядаються для взаємовідносин
3. За галузевою ознакою: внутрішньогалузева	Використання доступне лише у галузі телекомунікацій	Відсутній
4. Конкуренція за видами товарів: товарно-видова	Конкурують товари одного виду	Доробка функціоналу/простоти

5. За характером конкурентних переваг: не цінова	Унікальність, зручність та якість – основні характеристики, що конкурують	Непереривна розробка предметної області
6. За інтенсивністю: марочна	Надання товару імені та логотипу	Впровадження власної назви та власного знаку.

Таблиця 4.8 - Ступеневий аналіз конкуренції на ринку

4.3.5 Аналіз пропозиції

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари замітники
	Відсутні	Нові гравці	Підприємство	Власники застарілого обладнання	Дешевші аналоги
Висновки	Безпроблемний старт на ринку, отримання клієнтів	На старті конкурентів немає, є час підготуватись та проконтролювати ринок	Випуск товару здійснюється приватною фірмою	З часом клієнтів буде ставати все менше, є потреба у розробці інших напрямлень	Є шанс втрати позицій на ринку

Таблиця 4.9 - Аналіз конкуренції в галузі за М. Портером

Проаналізувавши дану інформацію, можна дійти висновку, що вихід на ринок можливий без значних перешкод. До сильних сторін, які має мати продукт є низька ринкова ціна, а також високий рівень відмовостійкості і легкості у використанні. Також слід додати, що має бути довгострокова підтримка продукту у разі виявлення проблем користувачем.

4.3.6 Аналіз та обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Ціна	Покупцю важливо затратити якнайменше ресурсів та отримати максимум результату
2	Функціонал	Має місце бути той факт, що покупцю важливіше буде конкретний функціонал, а ціна на товар вже другорядна характеристика

3	Репутація	Відгуки споживачів створюють обличчя надавача товару перед потенційними замовниками
4	Географічне розташування філіалів	Вкрай важливо покривати чимого більші території, на яких знаходяться потенційні покупці

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим						
			-3	-2	-1	0	+1	+2	+3
1	Ціна	6	-	-	+	-	-	-	-
2	Функціонал	7	-	-	+	-	-	-	-
3	Репутація	4	-	-	-	-	+	-	-
4	Географічне розташування	3	-	+	-	-	-	-	-

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін пристрою розирення функціоналу

4.3.7 SWOT - аналіз

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> – Високий функціонал; – Низька вартість; – Відсутність сильної конкуренції; 	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> – Відсутність репутації на старті; – Недостача коштів на створення кількох філіалів для роботи на ринках кількох держав.
<p>Можливості (O):</p> <ul style="list-style-type: none"> – Доробити пристрій аби зменшити вартість без втрати ефективності; 	<p>Загрози (T):</p> <ul style="list-style-type: none"> – конкуренти, які розробляють сдешевлені аналоги.

Таблиця 4.12 - SWOT аналіз стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
-------	--	--------------------------------	-------------------

1	Надання пристрою на обмежений проміжок часу	Середня	1 місяць
2	Презентація пристрою на галузенаправлених виставках	Висока	6-12 місяців

Таблиця 4.13 - Альтернативи ринкового впровадження стартап-проекту

З означених альтернатив, вважаю, кращою буде друга, бо ймовірність отримати ресурси вкрай висока, не взираючи на високий строк реалізації. Це обумовлено тим, що потенційні користувачі не захочуть одразу ж встановлювати обладнання на «тест-драйв», також існує вірогідність, що вони про нього можуть взагалі не дізнатися, окрім як з презентації.

4.4 Розроблення ринкової стратегії проекту

4.4.1 Розроблення ринкової стратегії проекту

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Власники застарілого телекомунікаційного обладнання	Вище середньої, бо проект є новим на ринку	80% сегменту ринку	Незначна інтенсивність	Легкість входу оцінюється вище середнього через відсутність сильних конкурентів.
Які цільові групи обрано: групу 1 (весь ринок)					

Таблиця 4.14. Вибір цільових груп потенційних споживачів

Виходячи з даного аналізу, приходимо до висновку, що підходяща цільова група – усі власники застарілого телекомунікаційного обладнання, що наразі працює. Тобто, можна визначити стратегію охоплення ринку, а саме – масовий маркетинг.

4.4.2 Формування стратегії розвитку та визначення стратегії конкурента

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Забезпечення збільшення маржі між собівартістю та ринковою ціною	Проведення контролю за витратами, зниження рекламних затрат.	Низька вартість; Позитивне відношення потенційних замовників	Стратегія лідерства по витратах

Таблиця 4.15 - Визначення базової стратегії розвитку

№ п/п	Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
1	Ні	Буде шукати нових споживачів	Ні	Стратегія лідера

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
	Максимальний функціонал	Атакуюча	Інноваційна і технологічна	Висока якість, відсутність потреби йти на компроміси
	Доступна ціна	Атакуюча	Ресурсна позиція	Ціна, «як для своїх»

Таблиця 4.17 - Визначення стратегії позиціонування

4.5 Розроблення маркетингової програми стартап-проекту

4.5.1 Формування маркетингової концепції та моделі

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Ефективність	Постійно високий рівень технологічності	Спрямування розробок, в першу чергу, на покращення роботи пристрою

Таблиця 4.18 - Визначення ключових переваг концепції потенційного товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Розширення функціоналу застарілих систем керування за рахунок зовнішнього модуля		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Вигляд та форма	М	Вр/Тх/Тл/Е
	2. Термін експлуатації	М	Вр /Тл
	3. Швидкодія	М	Вр/Тл
	Пакування: запакований у коробку для транспортування		
3. Товар із підкріпленням	До продажу: інструкція користувача		
	Після продажу: 24/7 підтримка		
Товар можна захистити за допомогою оформлення патенту як на саму модель роботи, так і на фізичний носій у сборі.			

Таблиця 4.19 - Опис трьох рівнів моделі товару

4.5.2 Формування системи збуту та розроблення комунікацій

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар
1	-	20000-25000\$	500000-1000000\$ в рік	Верхня – 21500 Нижня - 19500

Таблиця 4.20 - Визначення меж встановлення ціни

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	ЦГ – підприємці. То ж, зрозуміло, що вони будуть вимагати якомога меншу вартість.	Відправка пристрою разом зі спеціалістом, що встановить його, та проведе первинне налаштування	Нульовий канал	Працювати напряду з замовником, підтримувати постійний контакт

Таблиця 4.21 - Формування системи збуту

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Недовіра до нового гравця на ринку	Інтернет, мобільний зв'язок	Реклама, рекомендації серед ЦГ	Викликати довіру до товару	Надати продукт у користування на обмежений час збитковим підприємствам

Таблиця 4.22 - Концепція маркетингових комунікацій

Висновки до розділу

В заключенні можна виділити такі моменти:

- можливість ринкової комерціалізації наявна. Є попит, динаміка ринку для проекту позитивна, рентабельність на допустимому рівні;
- перспективи впровадження високі, так як бар'єр входження досить слабкий за рахунок відсутності серйозної конкуренції;
- альтернативним варіантом впровадження слід обрати презентацію пристрою на галузенаправлених виставках техніки;
- подальша імплементація доцільна

ЗАГАЛЬНІ ВИСНОВКИ

Дана магістерська дисертація присвячена розробці та побудові об'єктно-орієнтованої моделі системи управління телекомунікаційною мережею.

Розглянуто та описано базові системи управління, їх будову а також способи роботи. Проведено огляд аналогів, а також визначена постановка задачі, ціль та мета даної дисертації.

Здійснено введення в об'єктно-орієнтовану методологію розробки систем. Проведено об'єктно-орієнтований аналіз та проектування. Розкрито поняття інкапсуляції, модульності, ієрархії та інших. Виконано аналіз вимог і визначення специфікацій програмного забезпечення при об'єктному підході. Зазначений порядок виконання процесів.

Проведено побудову моделі системи управління телекомунікаційними мережами, яка дозволяє більш ефективно керувати мережею. Описано динамічні, статичні моделі та моделі реалізації об'єктно-орієнтованих програмних систем. Проведена розробка структури програмного забезпечення при об'єктному підході. Розкрито та описано поняття каналів та схем управління.

Створено ринкову (маркетингову) програму, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення.

Як результат даної дисертації було розроблено об'єктно-орієнтовану модель системи управління телекомунікаційною мережею.

ПЕРЕЛІК ПОСИЛАНЬ

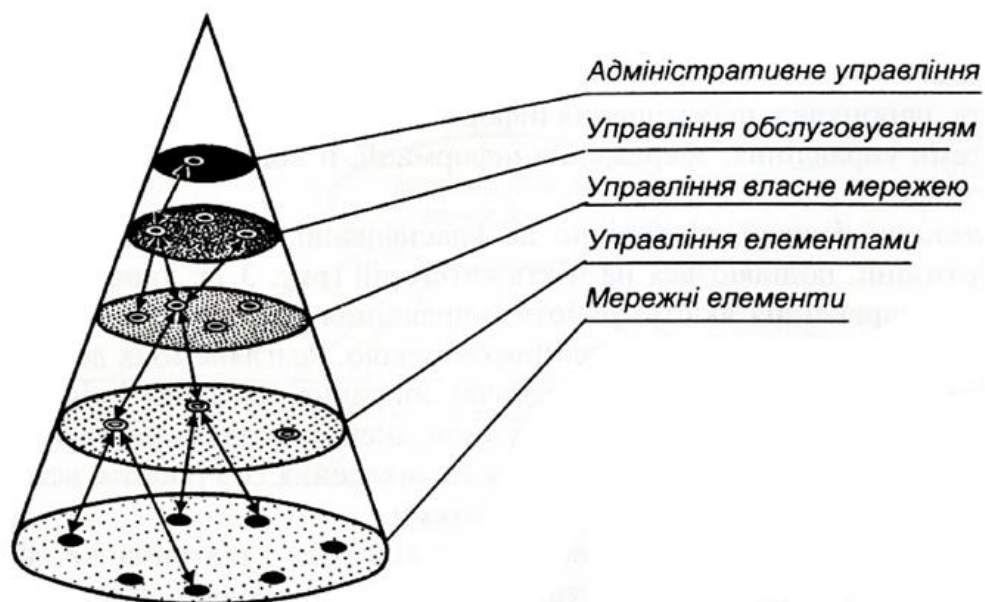
1. Об'єктно-орієнтований аналіз та проектування з прикладами додатків на C++ (Друге видання) – Г. Буч
2. Об'єктно-орієнтовані технології проектування прикладних програмних мереж – С.С. Гайсарян
3. Науковий журнал «Телекомунікаційні та інформаційні технології» [Електронний ресурс] – Режим доступу:
http://www.dut.edu.ua/uploads/p_130_11176947.pdf
4. Структура SNMP. Недоліки протоколу, та формат повідомлень [Електронний ресурс] – Режим доступу: https://studopedia.ru/11_82361_lektsiya-.html
5. Методи і засоби підвищення показників якості систем управління телекомунікаційними мережами – Л.Н. Беркман
6. Аналіз вимог і визначення специфікацій програмного забезпечення при об'єктному підході [Електронний ресурс] – Режим доступу: <http://um.co.ua/5/5-3/5-3943.html>
7. Поняття модульності [Електронний ресурс] – Режим доступу:
<http://um.co.ua/10/10-7/10-73718.html>
8. Проектування інформаційних систем. Навчальний посібник з курсу "Проектування інформаційних систем" [Електронний ресурс] – Режим доступу:
<https://studall.org/all2-114831.html>
9. Методи дискретної модуляції [Електронний ресурс] – Режим доступу:
<http://lecture.in.ua/metodi-diskretnoyi-modulyaciyi.html>
10. Стартап. Настольная книга основателя - С. Бланк, Б. Дорф; пер. с англ. Т. Гутман, И. Окунькова, Е. Бакушева. – 2-е изд. – Москва: Альпина Пабlishер, 2014. – 614 с.

ДОДАТКИ

ДОДАТОК А

Рівні управління мережею зв'язку

Рівні управління мережею зв'язку



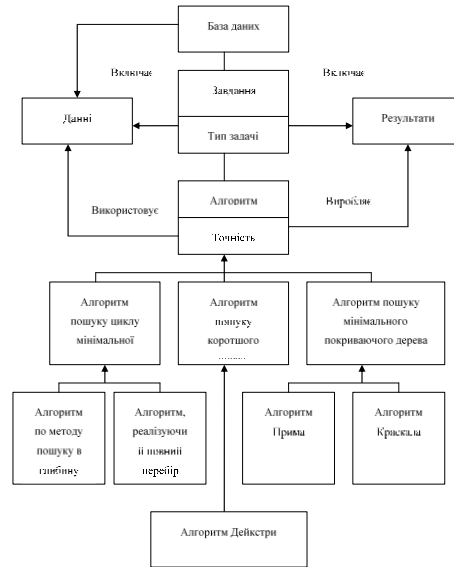
Демонстраційний плакат № 1
до магістерської дисертації на тему
„Об’єктно-орієнтована модель системи управління телекомунікаційною
мережею”

Розробив: Мошенченко М.С
Прийняв: Жураковський Б.Ю

ДОДАТОК Б

Контекстна діаграма класів для системи рішення комбінаторно-оптимізаційних
задач

Контекстна діаграма класів для системи рішення комбінаторно-оптимізаційних задач



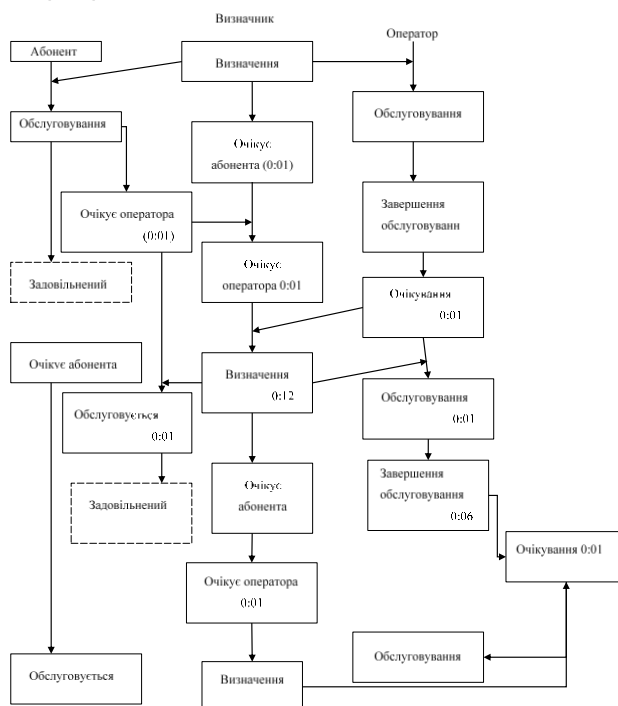
Демонстраційний плакат № 2
до магістерської дисертації на тему
„Об’єктно-орієнтована модель системи управління телекомунікаційною мережею”

Розробив: Мошенченко М.С
Прийняв: Жураковский Б.Ю

ДОДАТОК В

Схема каналу управління оператор – абонент

Схема каналу управління оператор-абонент

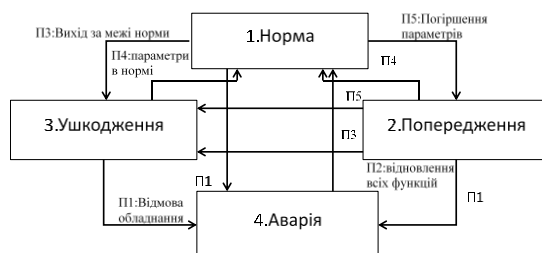


Демонстраційний плакат № 3
до магістерської дисертації на тему
„Об’єктно-орієнтована модель системи управління телекомунікаційною мережею”
Розробив: Мошенченко М.С
Приймав: Жураковський Б.Ю

ДОДАТОК Г

Діаграма переходів стану для об'єкту управління

Діаграма переходів стану для об'єкту управління



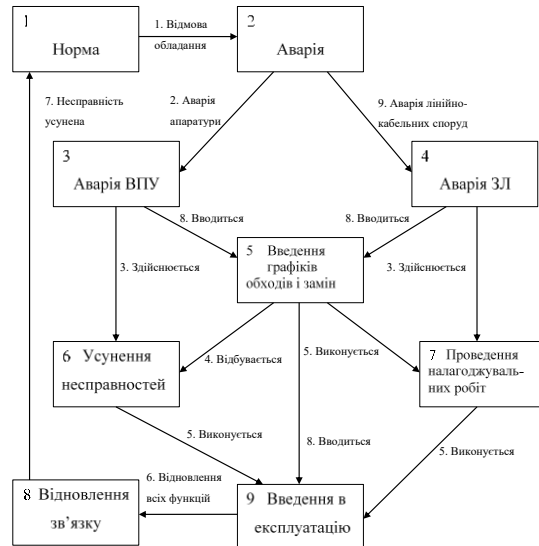
Демонстраційний плакат № 4
до магістерської дисертації на тему
„Об'єктно-орієнтована модель системи управління телекомунікаційною мережею”

Розробив: Моценченко М.С
Прийняв: Журавський Б.Ю

ДОДАТОК Д

Об'єктно-орієнтована модель системи управління телекомунікаційною мережею
при аварії вузлового пункту управління

Об'єктно-орієнтована модель системи управління телекомунікаційною мережею при аварії вузлового пункту управління



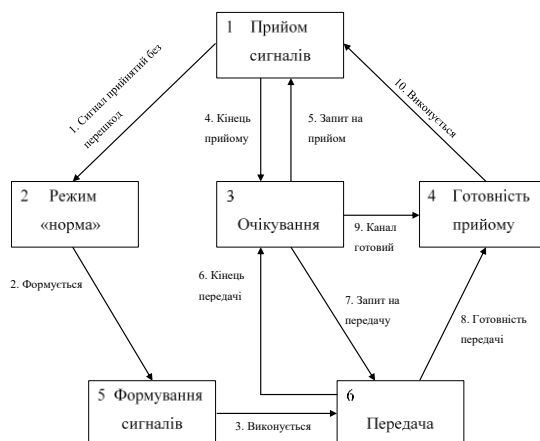
Демонстраційний плакат № 5
до магістерської дисертації на тему
„Об'єктно-орієнтована модель системи управління телекомунікаційною мережею”

Розробив: Мошенченко М.С
Прийняв: Жураковский Б.Ю

ДОДАТОК Е

Об'єктно- орієнтована модель системи управління телекомунікаційною мережею
в режимі «норма»

Об'єктно- орієнтована модель системи управління телекомунікаційною мережею в режимі «норма»



Демонстраційний плакат № 6
до магістерської дисертації на тему
„Об'єктно-орієнтована модель системи управління телекомунікаційною мережею”
Розробив: Мошенченко М.С
Прийняв: Жураковский Б.Ю